

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

**Realizace vzdáleně řízené úlohy určené pro  
virtuální laboratoř založené na moderních  
algoritmech řízení a technologii HTML5**

**Implementation of Remote Control in Virtual  
Laboratory Based on Modern Control Theory with  
the use of HTML5 Technology**

2013

Bc. Rudolf Vojtíšek

## Zadání diplomové práce

Student: **Bc. Rudolf Vojtíšek**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2601T004 Měřicí a řídicí technika

Téma: Realizace vzdáleně řízené úlohy určené pro virtuální laboratoř založené na moderních algoritmech řízení a technologii HTML5  
Implementation of Remote Control in Virtual Laboratory Based on Modern Control Theory with the use of HTML5 Technology

### Zásady pro vypracování:

1. Studium základních metod syntézy zpětnovazebních regulačních obvodů a studium metod moderní teorie řízení.
2. Návrh a sestavení fyzikálního modelu regulované soustavy určené pro virtuální laboratoř.
3. Analytická identifikace fyzikálního modelu.
4. Návrh regulátoru založeného na moderní teorii řízení pro zvolený fyzikální model a simulace regulačního obvodu v prostředí REX + Simulink.
5. Realizace navrženého algoritmu s využitím řídicího systému REX, zvolených V/V modulů a jednodeskového PC s OS Linux/Xenomai.
6. Vizualizace úlohy s využitím technologie HTML5.
7. Zhodnocení, závěr.

### Seznam doporučené odborné literatury:

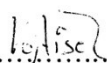
- [1] XUE, Dingyü, Yangquan CHEN a Derek P. ATHERTON. *Linear feedback control: analysis and design with MATLAB*. 4th rev. and enlarged ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, c2007. xii, 354 p. ISBN 08-987-1638-1/978-0-898716-38-2.
- [2] NOSKIEVIČ, Petr. *Modelování a identifikace systémů*. 1. vyd. Ostrava: MONTANEX, a. s., 1999. 276 s. ISBN 80-7225-030-2.
- [3] VÍTEČKOVÁ, Miluše a Antonín VÍTEČEK. *Základy automatické regulace*. 2. vyd. Ostrava: VŠB-TU Ostrava, 2008. 243 s. ISBN 978-80-248-1924-2.
- [4] VAVŘÍN, Petr. *Teorie automatického řízení I (Lineární spojité a diskrétní systémy)*. 2. přepracované vyd. Brno: VUT Brno, 1991. 158 s. ISBN 80-214-0244-X.
- [5] VAVŘÍN, Petr. *Teorie dynamických systémů*. 1. vyd. Brno: VUT Brno, 1989. 177 s.
- [6] ZÍTEK, Pavel, Milan HOFREITER a Jaroslav HLAVA. *Automatické řízení*. Vyd. 2., přeprac. Praha: Vydavatelství ČVUT, 1999. 148 s. ISBN 80-010-2044-4.
- [7] ŠULC, Bohumil. *Teorie automatického řízení s počítačovou podporou*. Vyd. 1. Praha: ČVUT, Strojní fakulta, 1999. 154 s. ISBN 80-010-1974-8.
- [8] OŽANA, Štěpán. *Navrhování a realizace regulátorů*. Ostrava: VŠB - TU Ostrava, 2012. Studijní materiály. ISBN 978-80-248-2605-9.
- [9] Firemní dokumentace Matlab and Simulink (MathWorks).
- [10] Firemní dokumentace Rex Controls.
- [11] Firemní dokumentace B-R Automation k programovatelným automatům a I/O modulům řady X20.

prof. RNDr. Václav Snášel, CSc.  
*děkan fakulty*

## **Prohlášení**

Prohlašuji, že jsem diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. 5. 2013

..........

## **Poděkování**

Rád bych na tomto místě poděkoval panu Ing. Štěpánu Ožanovi, Ph.D. za vedení mé diplomové práce a za cenné rady a konzultace a panu Ing. Martinu Piešovi, Ph.D. za pomoc při realizaci mého úkolu. Dále pak své rodině a v neposlední řadě své přítelkyni. Svým tolerantním přístupem a podporou mi vytvořili výborné podmínky pro studium a realizaci diplomové práce.

# Abstrakt

Diplomová práce se zabývá možnostmi regulace polohy cínové kuličky umístěné na dráze, která je vytvořena ze dvou napružených odporových drátů. Poloha kuličky je regulována pomocí průmyslového počítače ALIX. Počítač je přes USB rozhraní připojen k open-source vývojové platformě Arduino obsahující vstupně/výstupními piny. Jako akční člen regulačního obvodu je použit servomotor, kterým se řídí zdvih úhlu ramene, na kterém je mechanicky upevněna odporová dráha s pohybující se kuličkou. Všechny signály potřebné k uskutečnění regulace jsou svedeny do krabičky s obslužnou elektronikou. V diplomové práci je popsán a navrhnut regulátor pomocí metod moderní teorie řízení. Program pro regulaci fyzikálního modelu je vytvořen v řídicím systému REX a vizualizace je provozována ve webovém rozhraní využívající technologii HTML5.

Všechny komponenty použité pro regulaci polohy kuličky jsou umístěny na výukovém stojanu a společně tvoří jednotný model, který bude využíván studenty Katedry kybernetiky a biomedicínského inženýrství.

## Klíčová slova

Servomotor, ALIX, Arduino Uno, regulátor, regulační obvod, kulička na tyči, MATLAB/Simulink, REX, HTML5.

## Abstract

This thesis deals with the possibilities of positioning tin balls placed on the track, which is made up of two stretched resistive wires. Ball position is controlled by an industrial PC ALIX. The computer is connected via the USB interface to the open-source development platform Arduino containing the I/O pins. As the actuator control circuit is used servomotor, which governs the stroke angle of the arm on which it is mechanically fixed resistive track with the moving ball. All signals necessary to implement the regulation is drained into the box with the control electronics. The thesis describes a regulator designed using the methods of modern control theory. Program for controlling physical model is created in the control system REX and visualization is run in a web interface using HTML5 technology.

All components used for position control balls are placed on teaching rack and together form a unified model that will be used by students of the Department of cybernetics and biomedical engineering.

## Keywords

Servomotor, ALIX, Arduino Uno, Controller, control circuit, Ball&Beam Model, MATLAB/Simulink, REX, HTML5.

# Seznam použitých symbolů a zkratk

E	obraz regulační odchylky	
e	regulační odchylka	
G(s)	přenos v Laplaceově transformaci	
G <sub>R</sub>	přenos regulátoru	
G <sub>S</sub>	přenos soustavy	
IP	internetový protokol	
K <sub>p</sub>	zesílení proporcionální složky regulátoru	
OS	operační systém	
PID	proporcionálně-integračně-derivační regulátor	
PWM	pulse width modulation, pulzní šířková modulace	
s	nezávisle, proměnná u obrazu v Laplaceově transformaci	
TCP/IP	sada protokolů pro komunikaci v počítačové síti	
T <sub>d</sub>	derivační časová konstanta regulátoru	
T <sub>i</sub>	integrační časová konstanta regulátoru	
T <sub>k</sub>	perioda kmitů	[s]
u	akční veličina	
USB	universal serial bus, univerzální sériová sběrnice	
v	poruchová veličina	
w	žádaná veličina	
y	regulovaná veličina	

# Obsah

<b>1 Úvod</b>	<b>- 1 -</b>
<b>2 Modelářské servomotory</b>	<b>- 2 -</b>
2.1 Princip činnosti modelářského servomotoru	- 3 -
2.2 Způsob řízení servomotoru	- 4 -
2.2.1 PWM řízení	- 4 -
<b>3 Řídicí počítač ALIX</b>	<b>- 6 -</b>
3.1 Základní popis systémové desky ALIX	- 6 -
3.2 Oživení systémové desky ALIX	- 7 -
<b>4 Vývojová platforma Arduino</b>	<b>- 9 -</b>
4.1 Základní popis Arduina Uno	- 9 -
4.2 Připojení k PC a instalace ovladačů	- 10 -
4.3 Nastavení frekvence pro PWM výstupy	- 11 -
<b>5 Řídicí systém REX</b>	<b>- 13 -</b>
5.1 Struktura řídicího systému REX	- 13 -
5.2 Běh systému REX a diagnostika	- 14 -
5.2.1 RexCore	- 14 -
5.2.2 RexDraw	- 15 -
5.2.3 RexView	- 15 -
5.2.4 RexComp	- 15 -
5.3 Knihovna funkčních bloků	- 15 -
<b>6 Popis hardwarové části řešení</b>	<b>- 16 -</b>
6.1 Výukový stojan	- 17 -
6.2 Fyzikální model	- 18 -
6.3 Elektronika modelu	- 20 -
<b>7 Identifikace modelu</b>	<b>- 23 -</b>
7.1 Matematický popis modelu	- 23 -
7.2 Přepočítání úhlu ramene na úhel servomotoru	- 26 -
7.2.1 Přepočítání úhlů v MATLABu	- 28 -
7.3 Linearizace	- 29 -
<b>8 Návrh regulátoru</b>	<b>- 33 -</b>



8.1 Klasická teorie řízení .....	- 33 -
8.1.1 Návrh PID regulátoru .....	- 34 -
8.2 Moderní teorie řízení .....	- 35 -
8.2.1 Problematika optimálního lineárně kvadratického řízení.....	- 35 -
8.2.2 LQG řízení .....	- 36 -
8.2.3 Návrh LQG regulátoru.....	- 38 -
<b>9 Realizace regulátoru v řídicím systému REX.....</b>	<b>- 41 -</b>
<b>10 Vizualizace.....</b>	<b>- 44 -</b>
10.1 Virtuální laboratoř .....	- 44 -
10.2 Technologie HTML5.....	- 44 -
10.3 Vytvoření vizualizačního programu .....	- 45 -
10.3.1 Vizualizace simulace .....	- 47 -
10.3.2 Vizualizace reálného procesu .....	- 49 -
<b>11 Závěr.....</b>	<b>- 52 -</b>
<b>Literatura</b>	

# 1 Úvod

V automatizaci se pro řízení průmyslových zařízení nebo různých procesů nejčastěji využívají algoritmy regulátorů, které jsou implementovány do průmyslových počítačů a spolupracují s celou řadou snímačů.

Diplomová práce se zabývá regulací polohy cínové kuličky s využitím průmyslového počítače ALIX a vývojové platformy Arduino. Cílem práce je pomocí těchto komponent a s použitím řídicího software REX vytvořit regulační obvod. Na sestavení fyzikálního modelu je použit servomotor a odporová dráha, po které se pohybuje cínová kulička. Servomotor zajišťuje zvedání ramene odporové dráhy. Všechny řídicí komponenty jsou umístěny na DIN liště, která je připevněna k výukovému stojanu.

Druhá kapitola je zaměřena na popis servomotoru a možnosti jeho řízení. Popsáno je vnitřní složení motoru, zapojení řídicí elektroniky a princip řízení pomocí PWM modulace. Pro účely diplomové práce byl vybrán modelářský servomotor HS-311.

Třetí kapitola je věnována řídicímu počítači ALIX. Uvádí se jeho technické parametry a způsob oživení. ALIX byl vybrán záměrně díky jeho možnosti připojit se k internetu, kompaktním rozměrům a ceně. Pro účely diplomové práce je vhodným řešením.

Vývojová platforma Arduino je popsána ve čtvrté kapitole. Opět se uvádí technické parametry a základní popis. Arduino se pomocí USB kabelu připojuje k ALIXu a slouží pro zpracování vstupně/výstupních signálů. V kapitole je řešeno připojení k počítači a instalace důležitých ovladačů, bez kterých platforma nemůže pracovat.

Následující kapitola popisuje řídicí systém REX. V REXu se navrhuje regulační obvod, který se následně nahrává do jednodeskového počítače ALIX. V podkapitolách je uvedena struktura řídicího systému, běh systému, diagnostika a knihovna funkčních bloků RexLib.

Na hardwarovou část diplomové práce je zaměřena šestá kapitola. Zde je popsána komunikace mezi jednotlivými komponenty tvořící regulační obvod. Dále pak obrázek výukového stojanu, fyzikální model kuličky na tyči a jednotlivá schémata zapojení elektroniky. Vytvořená elektronika slouží pro upravování signálů dávající informaci o poloze kuličky na dráze a signálů řídicí otáčení servomotoru.

Šestá kapitola se věnuje analytické identifikaci regulované soustavy a problematice přepočtu závislosti úhlu zvednutí ramene na úhlu natočení servomotoru. Správný přepočet úhlů je nezbytný pro bezchybnou regulaci. Z matematicko-fyzikálního popisu kuličky na tyči je následně provedena linearizace v pracovním bodě.

Pro fyzikální model je pomocí moderní metody řízení navrhnut LQG regulátor a pomocí klasické teorie řízení navrhnut PID regulátor. Oba regulátory jsou namodelovány v Simulinku a regulují matematický model kuličky na tyči.

V deváté kapitole je čtenář seznámen s návrhem regulačního obvodu v samotném řídicím systému REX a jeho provozováním na cílové platformě.

Poslední kapitola se věnuje vizualizaci navržených schémat ve webovém prohlížeči. Vytvořené úlohy budou součástí virtuální laboratoře a bude je možno ovládat přes internet (a to buď jako simulaci nebo reálný proces). Je vysvětlen pojem virtuální laboratoř a výhody vzdáleně ovládaných experimentů. V této kapitole je popsáno vytvoření vizualizačních obrazovek a jejich obsluhu z pohledu uživatele. V grafech jsou zobrazeny průběhy regulace.

## 2 Modelářské servomotory

Modelářské servomotory (také označované jako serva) jsou určeny k ovládání rádiem řízených modelů. Mají široké využití v robotice, ovládání malých laboratorních a automatizačních zařízení nebo ovládání v domácnosti. Na servomotor je připojeno napájení a řídicí signál. V závislosti na řídicím signálu (*PWM*) dochází k pootočení výstupního hřídele o požadovaný úhel, který závisí i na kvalitě řídicí elektroniky. U kvalitních servomotorů je natočení velmi přesné. V diplomové práci je pro nastavení polohy použit analogový modelářský servomotor HS-311 od výrobce HITEC (Obr. 2).



Obr. 2: Servomotor HS-311

Obecně se modelářské servomotory skládají ze tří základních prvků:

- stejnosměrného motoru,
- převodovky,
- řídicí elektroniky.

Stejnoseměrný motor bývá nejčastěji v provedení s permanentními magnety. Tyto motory jsou hodně využívány v oblasti hračkářství a modelářství. Stator je tvořený permanentním magnetem a rotující kotva je ve formě elektromagnetu.

Převodovka slouží stejně jako u jiných motorů k nastavení parametrů síla/rychlost. Navíc ještě převádí velikost (úhel) natočení z motoru na zpětnovazební potenciometr. U analogového servomotoru HS-311 je plastové provedení převodových koleček.

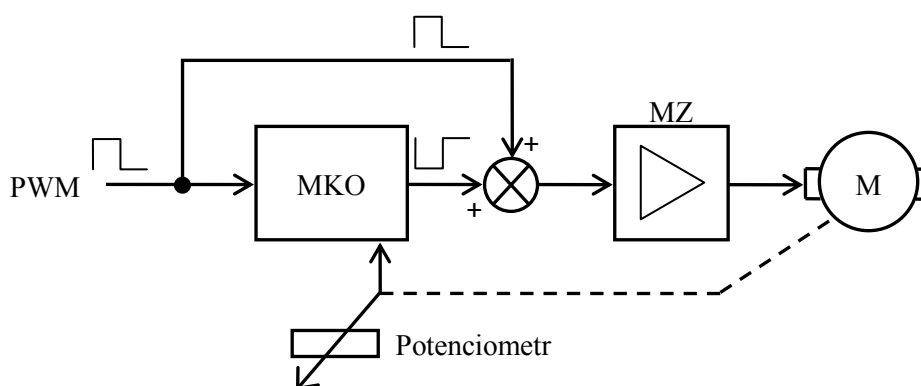
V řídicí elektronice se zpracovává vstupní *PWM* signál, ze kterého vyhodnocuje požadované natočení, které se porovnává s natočením zpětnovazebního potenciometru a natáčí motorem požadovaným směrem. Ideální je frekvence opakování pulzu vstupního signálu  $50\text{Hz}$ . Tyto signály jsou impulzy od  $1\text{ms}$  do  $2\text{ms}$ , pro natáčení od  $0^\circ$  do  $180^\circ$ . Natáčení servomotoru se provádí plynule.

Základní technické parametry modelářských servomotorů:

- rychlost – udávána výrobcem v  $[s/^{\circ}]$ ,
- tah (síla) – udávána výrobcem v  $[kg/cm]$ ,
- napájecí napětí – ve voltech  $[V]$ , na  $5V$  fungují všechny modelářské servomotory,
- klidový proud – udávaný výrobcem v  $[mA]$ ,
- maximální proud – maximální odebíraný proud při natáčení hřídele,
- typ převodovky – většinou plasty nebo kovové slitiny,
- velikost šasi – udává se v  $[mm]$ .

## 2.1 Princip činnosti modelářského servomotoru

Dnešní modelářské servomotory obsahují stejnosměrný motor s převodovkou a řídicí elektronikou. Zjednodušené zapojení elektroniky je vyobrazeno na následujícím obrázku (Obr. 2.1).



**Obr. 2.1: Blokové schéma zapojení řídicí elektroniky servomotoru**

Na vstup přichází řídicí *PWM* impuls, který spustí monostabilní klopný obvod (MKO). Ten vygeneruje impuls o délce odpovídající momentální poloze servomotoru. Vygenerovaný impulsní signál má opačnou polaritu než je vstupní řídicí impuls. Tyto dva impulsy se porovnávají a výsledkem je rozdílový impuls, který se zesiluje v můstkovém spínači (MZ). Zesílený signál se přenáší na svorky elektromotoru (M) a roztáčí ho jedním nebo druhým směrem.

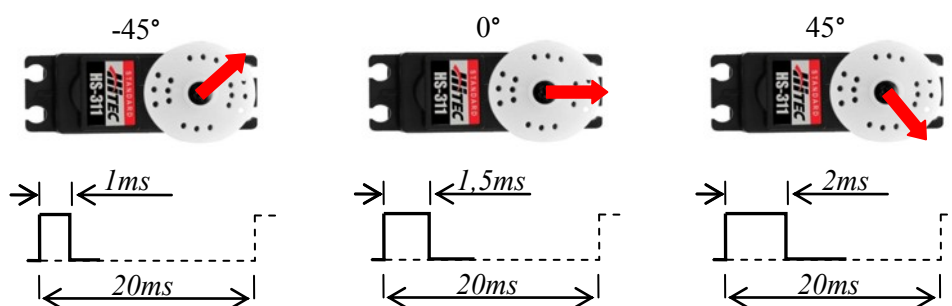
Elektromotor otáčí přes převodovku výstupní hřídeli a současně i potenciometrem, který působí jako zpětná vazba do monostabilního klopného obvodu. Směr otáčení je takový, že impuls generovaný monostabilním klopným obvodem se svojí délkou přibližuje délce vstupního *PWM* řídicího impulsu. Teprve až jsou oba impulsy stejně dlouhé, se elektromotor zastaví. V tomto

případě dosáhl servomotor polohu, která odpovídá v daném okamžiku přijímanému *PWM* řídicímu impulsu.

Servomotor je také možné upravit pro kontinuální otáčení odstraněním mechanického dorazu na převodovém kolečku. Tímto způsobem se získá levný a výkonný převodový motor s možností jednoduchého řízení směru a rychlosti otáčení.

## 2.2 Způsob řízení servomotoru

Servomotor je napájen a řízen po třech vodičích. Jako připojovací konektor je nejčastěji použit plochý trojpinový typ. Víme, že poloha natočení výstupní hřídele servomotoru odpovídá proporcionálně šířce řídicího impulsu. Řídicí impuls je pozitivní o velikosti amplitudy  $5V$  a rozsahu šířky proměnné od  $1ms$  do  $2ms$ . Takové šířce odpovídá rozsah polohy výstupní hřídele servomotoru o velikosti  $90^\circ$ . Tyto impulsy se v ideálním případě opakují s frekvencí  $50Hz$ . Rychlost opakování není kritická. Pokud budou řídicí impulsy přicházet s větším zpožděním, pouze se zpomalí pohyb servomotoru. Natačení hřídele v závislosti na velikosti řídicího impulsu je na Obr. 2.2.



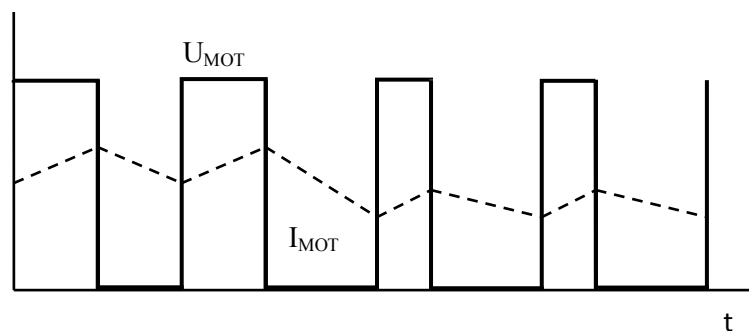
Obr. 2.2: Rozsah pohybu standardního servomotoru

Z předchozího obrázku je patrné, že nastavené šířce řídicího impulsu odpovídá rozsah polohy výstupní hřídele servomotoru  $90^\circ$ . Zvětšením tohoto rozsahu od  $0,5ms$  do  $2,5ms$  se mechanický rozsah pohybu natočení hřídele změní na  $180^\circ$ . Tento rozsah dovoluje nastavit většina modelářských servomotorů i v diplomové práci použitý typ HS-311.

### 2.2.1 PWM řízení

Princip spočívá v rychlém spínání a vypínání napájení. Díky setrvačnosti motoru a dostatečně vysoké frekvenci spínání, rotor nestačí změny sledovat. Otáčky motoru budou úměrné

střední hodnotě protékajícího proudu motorem  $I_{MOT}$ . Motor se chová, jako kdyby byl napájen napětím o velikosti střední hodnoty, která je dána poměrem doby zapnutí a vypnutí (Obr. 2.2.1).



**Obr. 2.2.1: Průběh proudu při PWM řízení [3]**

Frekvenci *PWM* signálu volíme tak, aby proud procházející motorem neklesal k nule. Rotor nestačí sledovat změny napětí v *PWM* signálu. Příliš nízká frekvence *PWM* signálu, kdy proud motoru klesá k nule, může zapříčinit zničení ložisek motorku. [3]

Pulzní-šířková modulace je výhodná, protože při ní nedochází, jako u lineární regulace k velkým výkonovým ztrátám na výkonových prvcích (převážně tranzistorech), které je nutné chladit. Je to také jediný kvalitní způsob řízení servomotorů.

## 3 Řídicí počítač ALIX

V diplomové práci se jako řídicí systém laboratorního modelu kuličky na tyči používá počítač ALIX spolu s vývojovou platformou Arduino, u které se využívají její vstupně/výstupní piny (bude popsána v následující kapitole).

ALIX je produkt společnosti PC Engines, která se zabývá vývojem systémových desek, redukcí IDE a jiných elektronických zařízení. PC Engines začal stavět rodiny desek, tak aby byly malé, neobsahovaly ventilátory, byly levné, ale měli stále silné servery. Jedná se o novou řadu platforem kompatibilních s procesory x86.

### 3.1 Základní popis systémové desky ALIX

Vzhledem k modulární architektuře ALIX-u lze jeho pomocí využít na více než 50 různých aplikacích v oblasti zabudovaných zařízení, bezpečnostních aplikací nebo pro směrování. Desky mohou být také osazeny v multimediálních aplikacích stejně jako na bezdrátových bodech nebo VPN bránách. Desky mají velice nízkou spotřebu, pohybující se od 3 W do 15 W.

Pro regulaci polohy kuličky na odporové dráze se používá základní desky verze ALIX.2D13. Základní specifikace desky je zobrazena v následující tabulce (Tab. 3.1).

Procesor	AMD Geode LX800, 500 MHz
Paměť DDR	256MB DRAM
Paměť pro OS a aplikace	CF karta v paměťovém slotu
Úložiště	CompactFlash patice, 44 pinová IDE lišta
Ethernet port	3x LAN 10/100, konektor RJ-45
Sériový port	1x RS-232C, konektor DB9
USB port	2x
MiniPCI slot	1x
CompactFlash slot	1x
Expanzní slot LPC	1x
FirmWare	tinyBIOS
Podporované OS	Velká řada, používáme Linux
Provozní teplota	0 ~ +50 °C
Provozní vlhkost vzduchu	Relativní vlhkost max. 70%
Napájení	Jack +7 ~ +20V <sub>DC</sub>
Spotřeba	3-4W bez PCI karet, celkem max. 15W
Rozměry (Š x V x H) (mm)	153 x 153 x 20
Hmotnost	300g

**Tab. 3.1: Technické parametry ALIX.2D13**

Rozmístění jednotlivých portů na desce ALIX z tabulky 3.1 je zobrazeno na Obr. 3.1.



Obr. 3.1: Základní deska ALIX.2D13

Tato deska je druhou verzí původní desky ALIX.2D3 a liší se tím, že má integrovanou RTC baterii.

## 3.2 Oživení systémové desky ALIX

Před uvedením ALIX-u do provozu bylo potřeba provést následující kroky:

- přivést napájení ze zdroje do napájecího konektoru,
- propojit ALIX s notebookem kříženým nebo nekříženým ethernetovým UTP kabelem. Pro propojení se vybere první zdířka vedle USB konektorů,



- na notebooku se nastaví vlastnosti sítě (tcp/ip), tak že budeme na stejném segmentu jako je ALIX (192.168.1.100). Tedy adresa 192.168.1.xxx,
- ověření propojení obou zařízení se provede v prohlížeči (Mozilla Firefox nebo v Google Chrome),
- v prohlížeči zadáme adresu <http://192.168.1.100> a přihlašovací údaje (Uživatel: root, Heslo: root),
- v zobrazeném grafickém prostředí pro ovládání ALIX-u se žádné další nastavení provádět nemusí a oživení je tedy kompletní,
- můžeme zavřít prohlížeč.

Počítač ALIX používaný pro účely diplomové práce, byl zakomponován do sítě na učebně E228, aby měl uživatel pohodlný přístupovat k výukovému stojanu pomocí školní internetové sítě (IP adresa: 158.196.151.57).

Stolní počítače nacházející se v učebně jsou k internetové síti připojeny síťovým ethernetovým kabelem a pouze přes ně je možné z webové vizualizace ovládat výukový stojan. Bezdrátový přístup k řídicímu PC se používá při nahrávání programu regulačního obvodu z notebooku. Webová vizualizace na notebooku je možná pouze v režimu simulace regulační úlohy (více v kapitole Vizualizace).

Tímto způsobem byl zvládnut první krok potřebný k vytvoření obslužného software pro diplomovou práci. Propojením osobního počítače a desky ALIX se dostáváme k dalšímu zařízení, které bude sloužit ke zpracovávání vstupně/výstupních signálů, a které je popsáno v následující kapitole.

## 4 Vývojová platforma Arduino

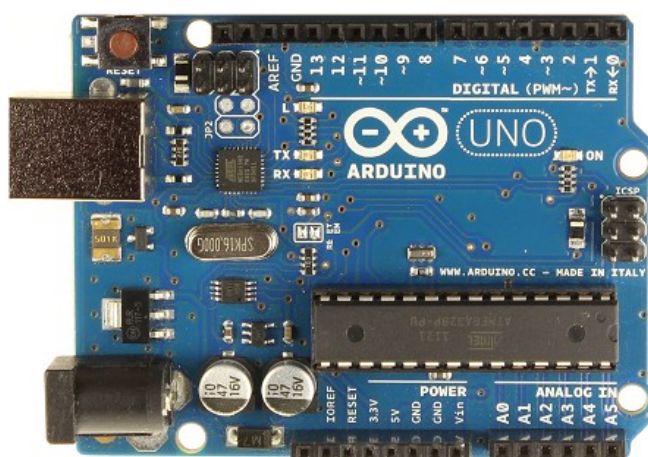
Jedná se o otevřenou softwarovou platformu založenou na mikrokontroleru ATmega a grafickém vývojovém prostředí. Arduino může být použito k vytvoření různých interaktivních zapojení nebo může být připojeno k softwaru na počítači. Připojení Arduino k počítači se využívá i u diplomové práce, kdy je Arduino přes USB kabel propojeno s počítačem ALIX.

Projekt Arduino je volně dostupný všem uživatelům, kteří jej chtějí volně používat nebo i vylepšovat. Lze samostatně složit nebo zakoupit už hotovou desku. V dnešní době je již několik verzí Arduino (Duemilanove, Uno, Due, Leonardo). V diplomové práci je použito Arduino Uno verze Rev3.

### 4.1 Základní popis Arduino Uno

Deska plošného spoje není příliš velká a zabírá plochu o velikosti 69 x 54 mm. Mozkem Arduino je osmi bitový mikrokontroler z rodiny AVR od firmy Atmel. Používané čipy jsou ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560. Verze Arduino Uno využívá čip ATmega328 taktovaný krystalem na frekvenci 16MHz. Tento čip je jako ostatní zmíněné uživatelsky programovatelný, má bootloader (kód, který se po spuštění postará o základní nastavení mikrokontroleru, jako je nastavení rozhraní USART, interní časovače a jiné) a nastavené potřebné fuses bajty (těmito bajty se nízkourovňově nastavují některé vlastnosti čipu). Díky těmto skutečnostem se uživatel nemusí starat o detaily a své programy píše v jazyce podobném C/C++.

Osazená deska plošného spoje Arduino Uno Rev3 je na následující obrázku (Obr. 4.1).



Obr. 4.1: Arduino Uno Rev3 [4]

Uživatel má k dispozici celkem 14 digitálních vstupně/výstupních pinů a 6 analogových vstupních pinů přístupných přes precizní patice, do kterých se připojují další obvody. Ze 14 digitálních pinů je 6 pinů možné použít na softwarově řízený PWM výstup.

Další nedílnou součástí Arduina je obvod zprostředkovávající komunikaci po USB rozhraní, pomocí kterého propojujeme desku s počítačem. Zde se konkrétně u naší verze využívá čip ATmega16U2. Samotné USB slouží i jako napájení nebo můžeme využít konektor pro 2,1mm jack. K propojení počítače a desky se používá USB kabel typu A/B.

Na desce jsou umístěny konektory pro ICSP (In Circuit Serial Programming) programování, resetovací tlačítko a několik LED diod. Nejdříve se po přivedení napájení rozsvítí LED s označením ON. Další dvě LED blikají při komunikaci s počítačem (RX- příjem dat, TX- odesílání dat). Poslední LED je ovládaná programem.

Technické specifikace Arduina Uno Rev3 jsou shrnuty v tabulce 4.1.

Mikrokontroler	ATMega328
Pracovní napětí	5 V <sub>DC</sub>
Doporučené vst. napětí	+7 ~ +12V <sub>DC</sub>
Limity vst. napětí	+6 ~ +20V <sub>DC</sub>
Digitální V/V piny	14 (z toho je 6 pinů s podporou PWM)
Analogové vst. piny	6
Proudové zatížení V/V pinů	40 mA
Flash paměť	32 kB (z toho 0,5 KB používá bootloader)
SRAM	2 kB
EEPROM	1 kB
Taktovací frekvence	16 MHz
Rozměry (Š x V) (mm)	69 x 54

**Tab. 4.1: Technická data Arduina Uno Rev3**

## 4.2 Připojení k PC a instalace ovladačů

Jak už bylo zmíněno výše, Arduino se připojuje k počítači pomocí USB kabelu typu A/B, kdy se konektor A připojí do počítače a konektor B do vývojové desky. Tímto způsobem byly komponenty propojeny a přechází se k instalaci ovladačů.

Z oficiálních stránek [www.arduino.cc](http://www.arduino.cc) se stáhne vývojové studio pro Arduino (verze používaná v diplomové práci je součástí přílohy IV). V našem případě byla zvolena verze pro Windows. Rozbalí se stažený soubor a spustí vývojové studio souborem Arduino.exe.

Nainstalování ovladačů na operačním systému Windows 7:

- Připojte Arduino k počítači a počkejte, než se začnou automaticky instalovat ovladače. Instalace ovladačů nebude však provedena správně a program po chvíli nahlásí chybu.
- Přejděte do složky „Ovládací panely“ a klikněte na „Systém“ a „Správce zařízení“.
- Ve „Správce zařízení“ otevřete nabídku „Porty“, kde by se měl zobrazit port s názvem Arduino UNO (COMxx).
- Klikněte pravým tlačítkem na tento port a zvolte „Aktualizovat software ovladače“ a následně vyberte „Vyhledat ovladač v počítači“.
- Ve složce, do které bylo nainstalováno vývojové studio, otevřete složku „drivers“ a v ní najdete soubor ovladače „Arduino UNO REV3.inf. Nevybírejte podsložku „FTDI USB Drivers“.
- Po těchto krocích dokončí Windows instalaci úspěšně.

Nyní uživatel spustí vývojový program Arduino IDE, kde nastaví typ správné vývojové desky a další nezbytně důležité parametry pro správnou funkci zařízení:

- v menu zvolíme hlavičku „Tools“, záložku „Board“ a vybereme „Arduino UNO“,
- ve stejné hlavičce zvolíme záložku „Seriál Port“ a vybereme správný port, na kterém jsou zařízení (počítač a Arduino) propojena,
- opět ve stejné hlavičce zvolíme záložku „Programmer“ a vybereme „USBasp“.

Pro návrh regulačního obvodu se používá řídicí systém REX, který je popsán v následující kapitole. Důležitou skutečností je kompatibilita řídicího systému REX a vývojové desky Arduino a to díky příkazům a funkcím, které byly vytvořeny vývojáři z REX Controls s.r.o. Tyto soubory jsou dodány v příloze IV.

Regulační algoritmus se všemi soubory musí být vložen do složky s názvem „REXduino“, jinak neproběhne správná kompilace a nahrání programu do Arduina.

## 4.3 Nastavení frekvence pro PWM výstupy

Tato podkapitola je zaměřena na nastavení frekvence *PWM* výstupů Arduina. Správné nastavení frekvence má vliv na kontinuální otáčení hřídele servomotoru. Pokud by toto nebylo provedeno, v servomotoru by docházelo k sekání hřídele ze strany na stranu a tím by nebylo možné regulovanou soustavu uvést do klidového stavu.

Nastavení frekvence je možné pouze přímo v programu pro Arduino. Postupuje se následovně:

- na konec souboru REXduino\_functions musí být přidána funkce setPwmFrequency() ze stránek <http://playground.arduino.cc/Code/PwmFrequency>, kód uvedený na stránkách je vidět níže:

```
void setPwmFrequency(int pin, int divisor) {
  byte mode;
  if(pin == 5 || pin == 6 || pin == 9 || pin == 10) {
    switch(divisor) {
      case 1: mode = 0x01; break;
      case 8: mode = 0x02; break;
      case 64: mode = 0x03; break;
      case 256: mode = 0x04; break;
      case 1024: mode = 0x05; break;
      default: return;
    }
    if(pin == 5 || pin == 6) {
      TCCR0B = TCCR0B & 0b11111000 | mode;
    } else {
      TCCR1B = TCCR1B & 0b11111000 | mode;
    }
  } else if(pin == 3 || pin == 11) {
    switch(divisor) {
      case 1: mode = 0x01; break;
      case 8: mode = 0x02; break;
      case 32: mode = 0x03; break;
      case 64: mode = 0x04; break;
      case 128: mode = 0x05; break;
      case 256: mode = 0x06; break;
      case 1024: mode = 0x07; break;
      default: return;
    }
    TCCR2B = TCCR2B & 0b11111000 | mode;
  }
}
```

[4]

- následuje krok, kdy do void setup() v REXduino musí být přidán příkaz setPwmFrequency(5, 1024).

Těmito kroky se nastaví frekvence *PWM* na pinu 5, na kterém je hodnota 62500Hz. Číslo 1024 označuje dělitele frekvence. Výsledkem je tedy frekvence daná vztahem (1):

$$f = 62500\text{Hz}/1024 = 61\text{Hz} \quad (1)$$

Frekvence *PWM* pinu 5 je pomocí vloženého kódu nastavena na hodnotu 61Hz.

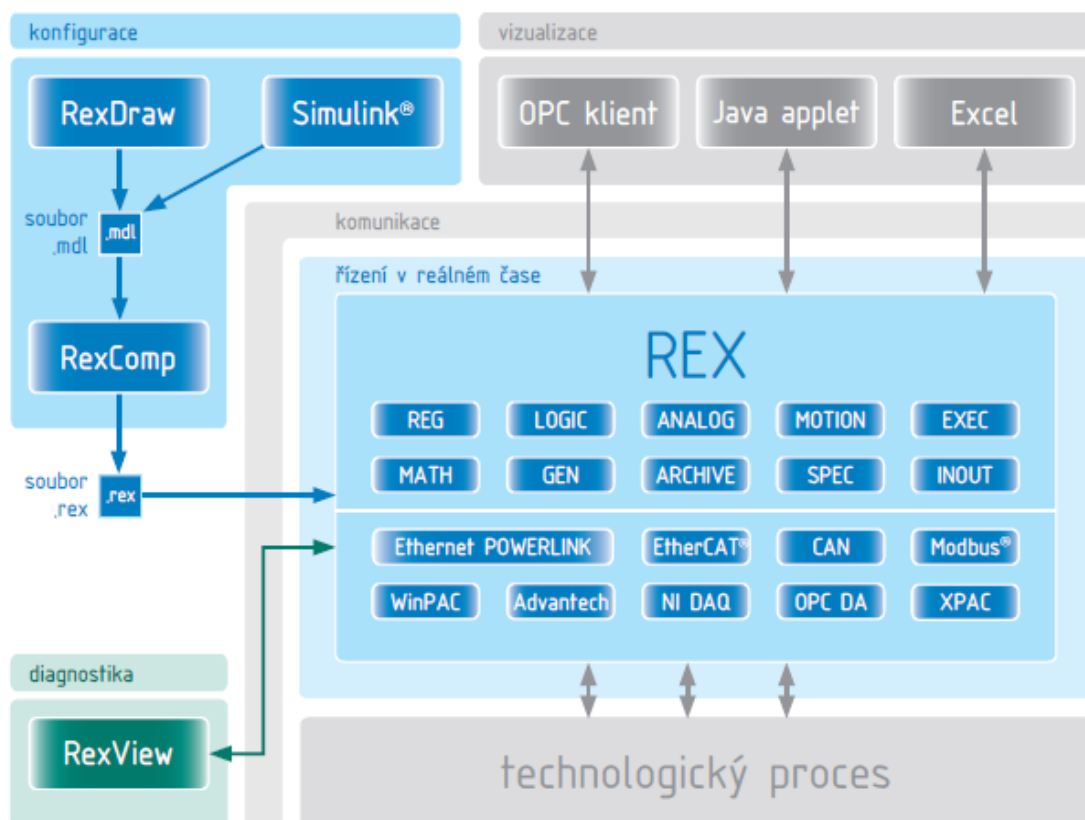
## 5 Řídicí systém REX

Jedná se o multiplatformový řídicí systém pro návrh a regulaci komplexních algoritmů automatického řízení. Výsledný algoritmus je sestaven z knihoven funkčních bloků, které pokrývají oblasti automatizace, regulace a nabízejí i řadu prvků pro tvorbu pokročilých algoritmů řízení. Pro speciální aplikace je možné zařadit vlastní funkce za pomoci volně programovatelného bloku. REX má základní myšlenkou kompatibilitu s programovým balíkem MATLAB/Simulink. Je implementován na platformě standardních operačních systémů Windows ( NT/2000/XP/Vista), Windows CE a operačního systému reálného času PharLap ETS. [3]

Výhodou systému REX je možnost návrhu řídicího algoritmu v prostředí MATLAB/Simulink, ve kterém se může i simulovat. Vlastní návrhy v REX-u jsou vytvářeny v jeho návrhovém prostředí RexDraw. Licence pro MATLAB/Simulink není nutná. [3]

### 5.1 Struktura řídicího systému REX

Blokově popsaná struktura řídicího systému REX je na následujícím obrázku (Obr. 5.1).



Obr. 5.1: Struktura řídicího systému REX [5]

Výhodou systému REX je, že se může navrhovat řídicí schémata v MATLAB/Simulinku nebo v programu RexDraw, který je součástí instalačního balíčku. Jedním z těchto zmíněných způsobů je vytvoření řídicího schématu v podobě souboru s koncovkou *.mdl*. Soubor se následně přeloží (zkompiluje) překladačem RexComp do souboru *.rex*. Tento soubor je možné buď přes diagnostický program RexView nebo přímo přes Simulink nahrát do jádra systému REX (RexCore). Po spuštění běží RexCore na cílovém zařízení, přičemž vykonává jednotlivé činnosti související s během řídicích schémat.

Vizualizační nástroj (např. Promotic) pak komunikuje s jádrem RexCore přes OPC server. Historické trendy i archivy se přes rozhraní Automation mohou vyčítat například do Excelu.

Podporované platformy: [5]

- Windows XP / Vista / 7
- Windows 2009 Embedded / XP Embedded
- Windows CE
- GNU / Linux
- GNU / Linux + Xenomai
- PharLap ETS

Komunikace a sběr dat: [5]

- Ethernet POWERLINK
- EtherCAT
- Modbus TCP/IP, Modbus RTU
- CAN
- OPC Data Access 2.0 a 3.0
- RS-232
- Zásuvné nebo USB I/O karty a moduly

Systém REX se využívá v hardwarech WinPAC, IPC, MOXA nebo i na notebookech. V diplomové práci je použit v řídicím počítači ALIX pracujícím na operačním systému GNU Linux (+ Xenomai), jako řídicí systém se vzdáleným sběrem dat, *1kHz*.

## 5.2 Běh systému REX a diagnostika

### 5.2.1 RexCore

Jedná se o jádro řídicího systému REX, které běží trvale na cílovém zařízení (ALIX). Jádro je komplexní program provádějící paralelně různé činnosti. Jednotlivé úlohy se vykonávají na základě priorit v režimu preemptivního multitaskingu pomocí několika subsystémů jádra.

### 5.2.2 RexDraw

V programu RexDraw probíhá návrh funkčních schémat řídicího systému REX. Schémata jsou tvořena bloky z knihovny systému REX. Všechny bloky pracují v diskretním čase, část je diskretizována pro danou periodu vzorkování. Vytvořený soubor s funkčním schématem má příponu *.mdl*. [3]

### 5.2.3 RexView

Program RexView umožňuje sledovat, co se děje v jádře řídicího systému REX při jeho běhu, a proto je velmi důležitým nástrojem při uvádění řídicího systému do provozu i v případě vzniku nějakých problémů během rutinního provozu. Program poskytuje detailní hierarchicky uspořádané informace o všech subsystémech jádra. Komunikace pomocí protokolu TCP/IP umožňuje připojit se k běžícímu jádru na lokálním počítači, v lokální síti i ve vzdálené síti (např. přes Internet). [5]

### 5.2.4 RexComp

Na základě hlavního souboru projektu aplikace ve formátu *.mdl* generuje program RexComp binární konfigurační soubor *.rex* řídicího systému REX. Při svém spuštění vypisuje překladač informace o překládaných souborech a případně i výskyt chyb překladu. Pokud je detekována závažná chyba, ukončí se překlad konfigurační soubor a nevytvoří se výsledný binární soubor. Překladač je možné spouštět přímo z konfiguračního programu RexDraw v menu Compiler/Compile. [5]

## 5.3 Knihovna funkčních bloků

Řídicí systém REX obsahuje rozsáhlou knihovnu funkčních bloků RexLib, v současné době tvořené více než 200 bloky určených pro průmyslové řízení technologických procesů a strojů:

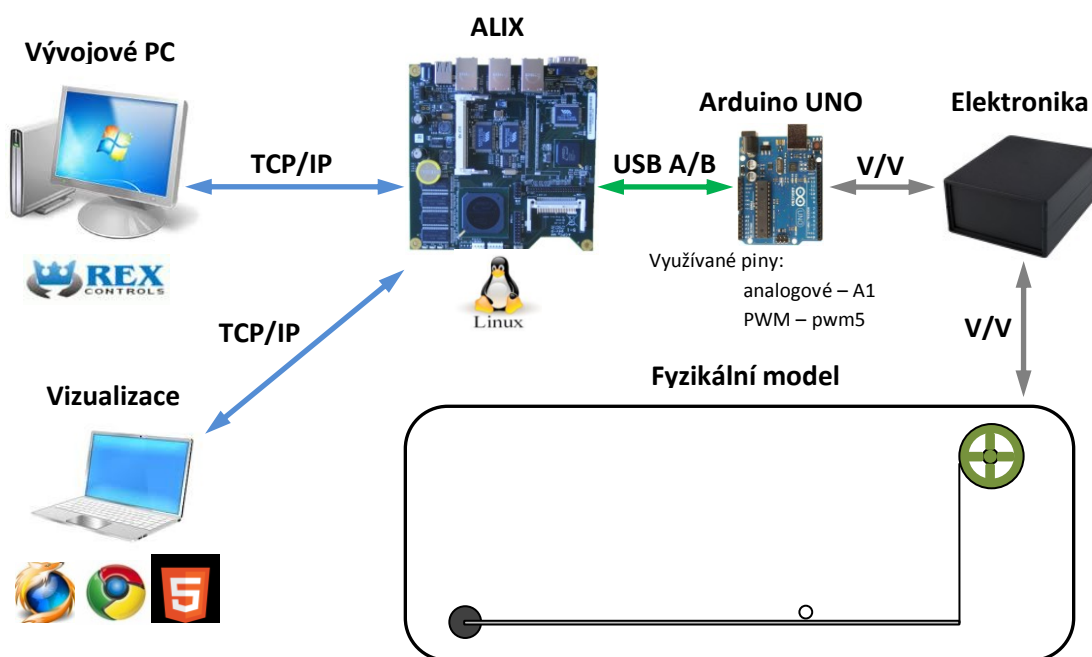
- EXEC – Konfigurace exekutivy reálného času
- INOUT – Bloky vstupů a výstupů systému REX
- MATH – Matematické bloky
- ANAL – Zpracování analogového signálu
- REG – Bloky pro regulaci
- GEN – Generátory signálu
- LOGIC – Logické řízení
- ARC – Archivace dat
- SPEC – Speciální bloky



## 6 Popis hardwarové části řešení

Pro regulaci polohy kuličky na odporové dráze, se využívá několik systémů vzájemně propojených tak, aby uživatel mohl pohodlně ovládat model například přes školní internetovou síť. K těmto systémům je vytvořena deska plošného spoje s elektronikou upravující vstupně/výstupní signály, které regulují otáčení servomotoru a dávají informaci o poloze kuličky na odporové dráze. Všechny komponenty jsou umístěny na DIN liště připevněné k výukovému stojanu. Fyzikální model se k elektronice připojuje přes svorky rovněž umístěné na DIN liště.

Propojení jednotlivých systémů pomocí jejich komunikačních rozhraní je zobrazeno na následujícím obrázku (Obr. 6).

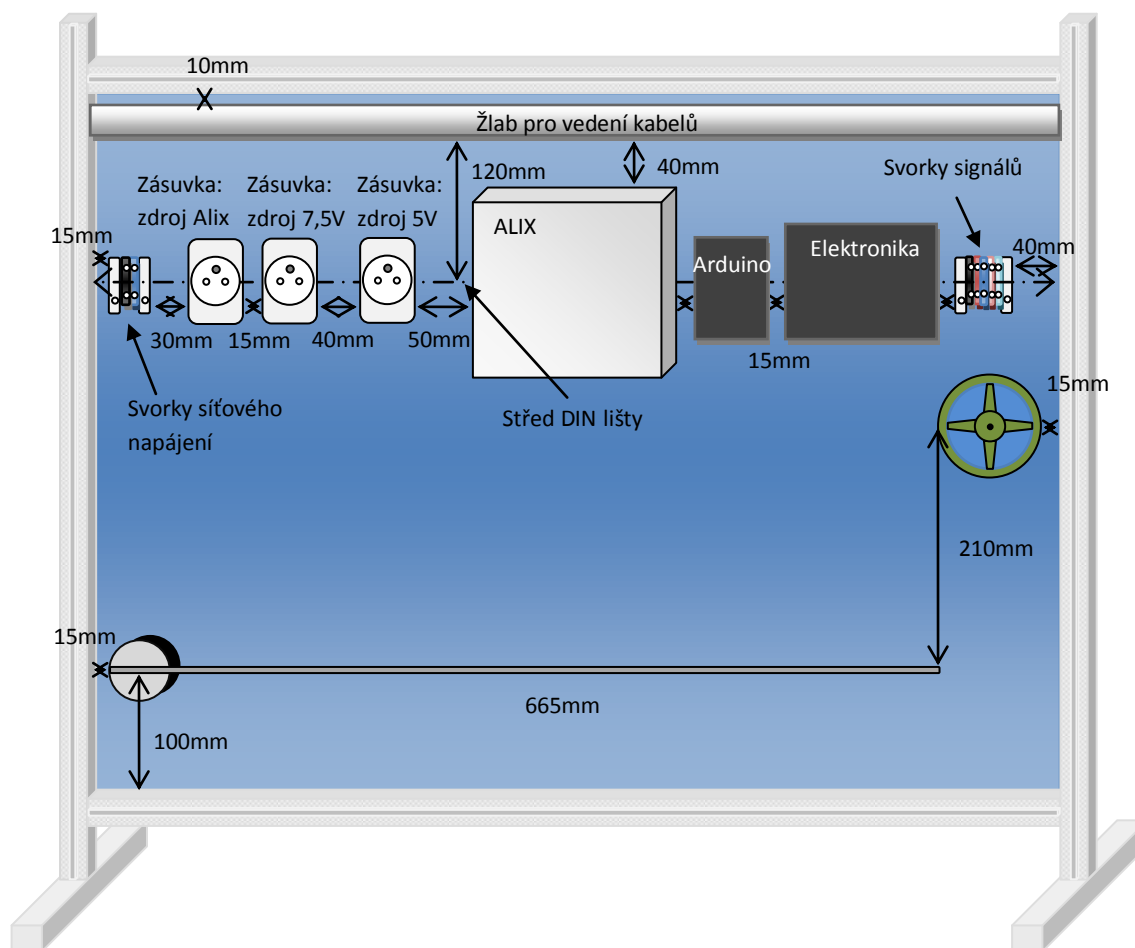


Obr. 6: Blokové schéma zapojení a komunikace

Na vývojovém počítači je v programu REX vytvořena řídicí aplikace, které se nahraje přes ethernetový kabel (případně přes Wifi síť v rámci školní sítě) do řídicího počítače ALIX. ALIX pracuje na operačním systému Linux a je zařízením, které provádí samotnou regulaci. Přes USB kabel předává a přijímá informace z/do vývojové platformy Arduino. V Arduino musí být nahrány ovladače zajišťující kompatibilitu se systémem REX (bylo popsáno v kapitole 4). Na analogový vstup Arduino je přiveden z krabičky elektroniky zesílený signál dávající informace o poloze kuličky. PWM signál z Arduino je opět přes krabičku elektroniky upraven a přiveden na vstupní svorky servomotoru. Celý model bude možné ovládat přes internet z webového rozhraní využívající technologie HTML5.

## 6.1 Výukový stojan

Stěžejním bodem diplomové práce bylo vytvořit fyzikální model kuličky pohybující se po odporové dráze. Tento model bylo nutné umístit na výukový stojan a doplnit ho veškerými zařízeními a prvky potřebnými k uskutečnění regulace. Obrázek návrhu rozmístění zařízení a prvků na výukovém stojanu je vidět níže (Obr. 6.1). Na obrázku jsou také uvedeny rozměry fyzikálního modelu a odsazení jednotlivých zařízení od sebe.



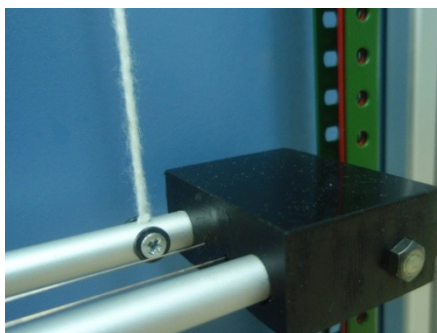
Obr. 6.1: Výukový stojan s fyzikálním modelem kuličky na tyči

Veškeré zařízení a doplňující prvky byly umístěny na DIN lištu tak, aby byly vizuálně osamostatněny a nepřekáželi modelu v jeho činnosti. Rameno s odporovou dráhou je drženo ve vodorovné poloze pomocí provázku, který se při regulaci navíjí na kolečko servomotoru. To sebou přináší problém v podobě rozdílných úhlů natáčení ramene a servomotoru. Pro správnou činnost regulace je nutné tyto úhly přepočítat a tím zjistit o kolik se musí natočit servomotor, aby se rameno zvedlo o požadovaný úhel viz. kapitola Identifikace modelu.

## 6.2 Fyzikální model

Fyzikálním modelem se v tomto případě rozumí pouze část výukového stojanu, kde je umístěna odporová dráha a kolečko servomotoru, na které se navíjí provázek zvedající rameno odporové dráhy. V této kapitole je popsáno vytvoření odporové dráhy a spojení s kolečkem servomotoru.

Odporová dráha se skládá z ložiska, kostry a dvou odporových drátů, které jsou napruženy tak, aby se po nich mohla pohybovat cínová kulička, která s odporovou dráhou představuje regulovanou soustavu. Pro vytvoření odporové dráhy se použily odporové dráty z materiálu Manganin. Výrobce udává elektrický odpor  $3,465\Omega/m$ , průřez  $0,4mm$ . Dráty musí být od sebe navzájem izolovány, protože se v obvodu používají pro měření odporu. Jejich spojení by znamenalo znehodnocení celého měření a regulace by za této podmínky nemohla fungovat. Detail odporové dráhy je zobrazen na následujícím obrázku (Obr. 6.2a).



**Obr. 6.2a: Odporová dráha s dorazem**

Na jeden drát je přivedeno vstupní napětí, které se přes kuličku a druhý odporový drát přenáší na vstup do elektroniky, která má za úkol tento signál zesílit a přenést ho na analogový vstup Arduina. Kompletní odporová dráha je mechanicky připevněna ke kostře a ta pak k ložisku, které je pomocí šroubů upevněno k výukovému panelu (Obr. 6.2b).



**Obr. 6.2b: Rameno odporové dráhy**

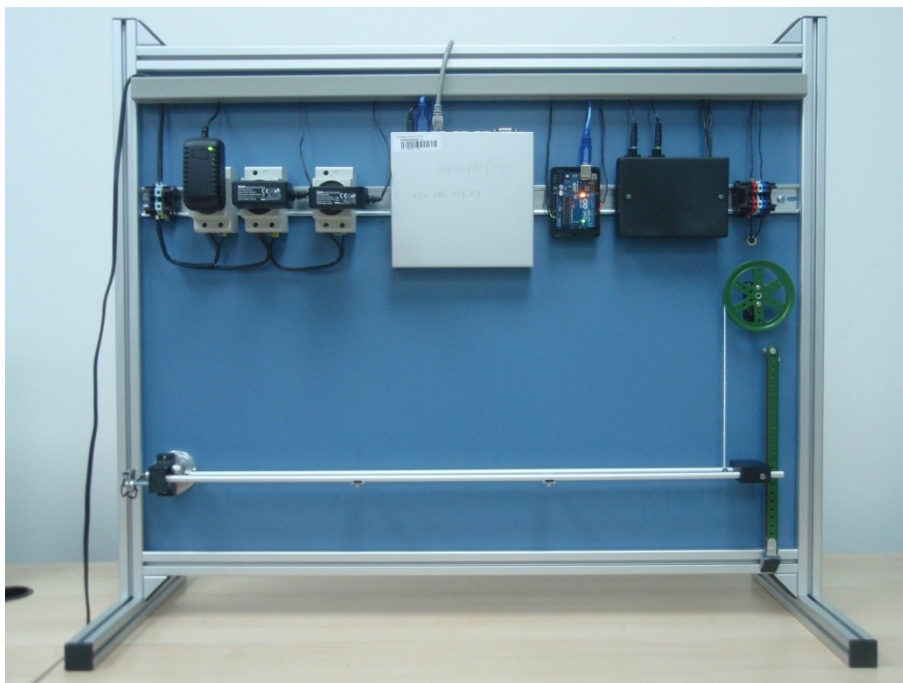
Na hřídeli servomotoru je šroubky pevně uchyceno kolečko, na které se při otáčení motorku navíjí provázek, který zvedá rameno odporové dráhy. Servomotor HS-311 je opět pomocí šroubků upevněn k výukovému panelu. Všechny signály zajišťující chod servomotoru jsou přes svorkovnici, umístěné na DIN liště, svedeny do plochého konektoru, který je součástí servomotoru.

Na Obr. 6.2c je vidět akční člen tak, jak je na stojanu umístěn.



**Obr. 6.2c: Servomotor HS-311 s kolečkem pro navíjení provázku**

Výukový stojan s kompletním fyzikálním modelem kuličky na tyči je vidět na Obr. 6.2d.



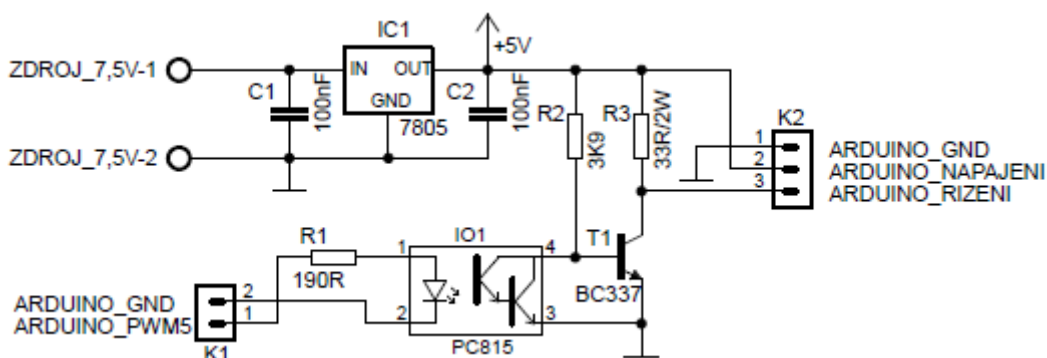
**Obr. 6.2d: Výukový stojan**

## 6.3 Elektronika modelu

Elektronika modelu je složena z části pro servomotor a z části pro měření na odporové dráze. Schémata zapojení jednotlivých částí jsou v textu zobrazeny níže. K vytvoření desky plošného spoje byla použita jednostranná cuprexitová deska. Seznam součástek použitých k osazení desky plošného spoje je uveden v příloze I.

Část elektroniky pro servomotor obsahuje optočlen PC815, na který je přiváděn *PWM* signál z Arduina. Optočlen byl zvolen záměrně kvůli galvanickému oddělení Arduina od jiných částí elektroniky. To má význam hlavně u poruchových stavů, kdy může dojít ke spálení elektronických obvodů.

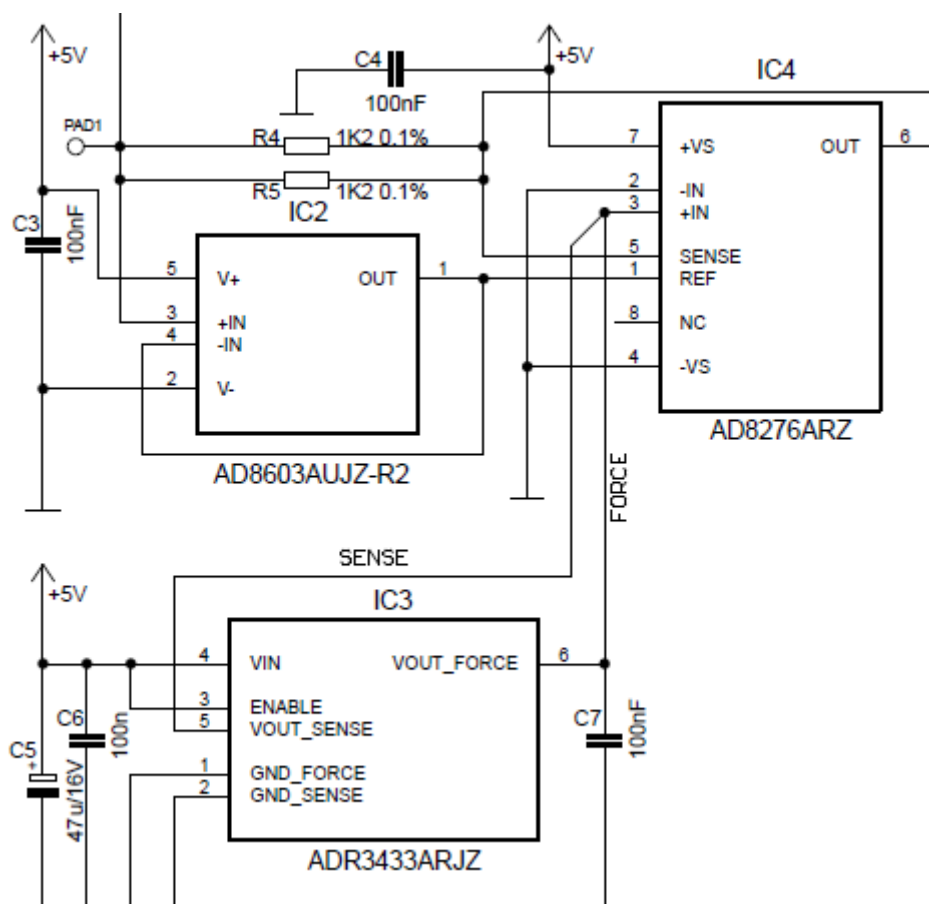
Servomotor HS-311 je napájen stejnosměrným napětím  $5V$  přes stabilizátor 7805 a je řízen *PWM* signálem přes bipolární tranzistor BC337. Pro přivedení signálů na servomotor byl použit konektor se zámkem do DPS se třemi kontakty. Stejný typ konektoru (nyní pouze se dvěma kontakty) byl použit pro přivedení *PWM* signálu do desky z Arduina. Těmito konektory také zabráníme případnému přepólování při další práci s modelem. Zdroj  $7,5V$  je umístěn mimo plastovou krabičku. Schéma zapojení elektroniky pro řízení otáčení servomotoru je na Obr. 6.3a.



Obr. 6.3a: Řízení servomotoru

Následující schéma zapojení je rozděleno na dvě části a slouží pro zpracování signálu z odporové dráhy. Opět byly použity konektory se zámkem do DPS, na které se připojují dva odporové dráty a je vyveden upravený signál na analogový vstup Arduina.

Na první části schématu zapojení (Obr. 6.3b) je vidět přesný proudový zdroj tvořený obvodem AD8276, AD8603 a ADR3433. Tento proudový zdroj dokáže dodávat konstantní proud nižší než  $15mA$ . V případě tohoto zapojení je dodáván proud o velikosti  $5,5mA$ , který se nastavuje přes odpory R3 a R4. Na pájecí bod PAD1 se připojuje propojka k propojení proudového zdroje s částí elektroniky, která slouží pro upravení signálu na hodnotu vhodnou pro analogový vstup Arduina.



Obr. 6.3b: Proudový zdroj

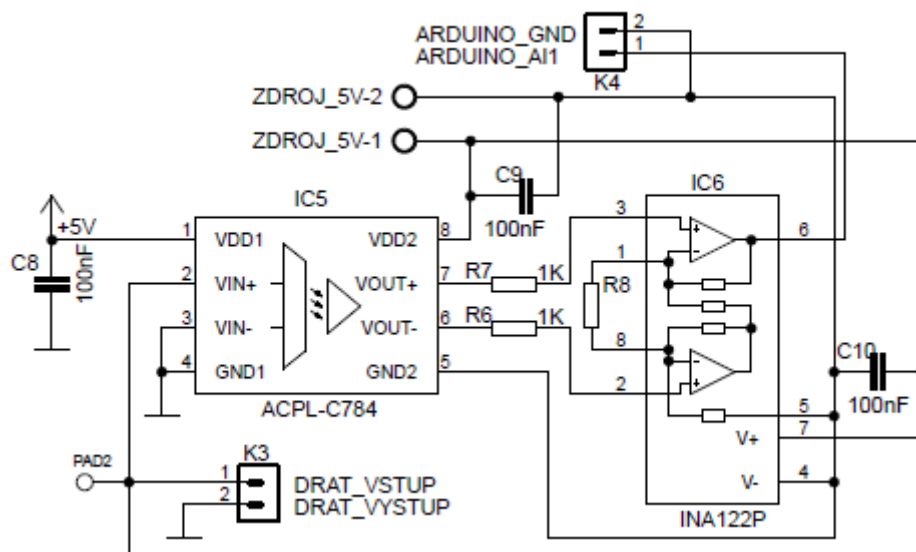
Arduino Uno má společný zemnicí vodič pro analogové vstupy i *PWM* výstupy a to vede k problému. Kdyby byl totiž signál z odporových drátů pouze zesílen a přiveden na analogový vstup Arduino, tak by se optočen PC815 stal nefunkčním. V tomto případě by galvanické oddělení Arduino od zbytku elektroniky v krabici nefungovalo, protože se pro měření využívá zemnicí vodič napájecího zdroje 7,5V. Proto musí být signál z odporových drátů nejprve přiveden na oddělovací zesilovač ACPL-C784, který má udávané vnitřní zesílení 8. Tento zesilovač má galvanicky oddělené země a tímto řeší vzniklý problém. Na jeho výstupu je připojen přesný přístrojový zesilovač INA122 s odporem R8 zvoleným tak, aby na výstupu přístrojového zesilovače byl původní měřený signál zesílen na hodnotu 0-5V pro analogový vstup Arduino. Odpor byl zvolen o velikosti 100kΩ a výsledné zesílení bude dáno vztahem (2).

Výpočet zesílení:

$$G = 5 + \frac{200k\Omega}{R8} = 5 + \frac{200k\Omega}{100k\Omega} = 7 \quad (2)$$

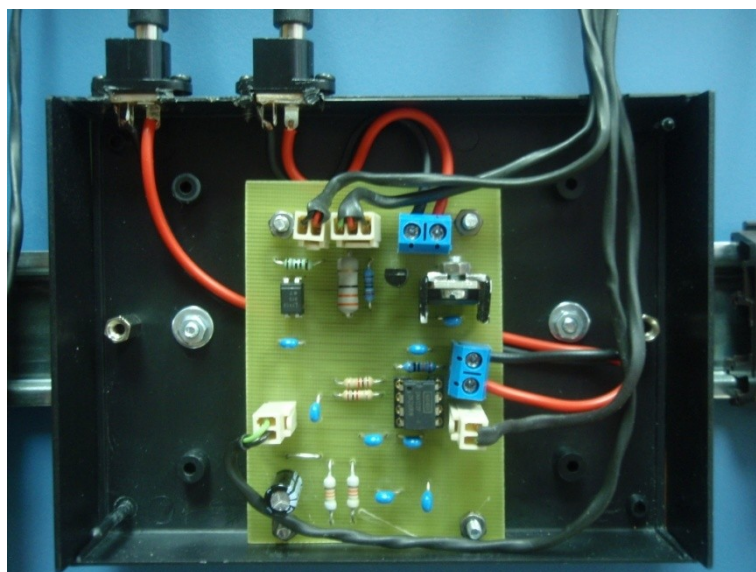
Schéma zapojení části elektroniky se zesilovači je na Obr. 6.3c.





**Obr. 6.3c: Zapojení zesilovačů**

Plastová krabička obsahující zapojení elektroniky z obrázků výše je zobrazena na následujícím obrázku (Obr. 6.3d). Na krabičce jsou k napájecím konektorům připojeny zdroje 5 V a 7,5 V. Dále jde vidět vyvedení všech signálů a umístění krabičky na DIN liště výukového stojanu.



**Obr. 6.3d: Plastová krabička s deskou elektroniky**

Deska plošného spoje byla vytvořena v programu Eagle verze 6.4.0 a je součástí přílohy I.

## 7 Identifikace modelu

Je důležité, aby soustava, která bude navrženým regulátorem regulována, byla správně identifikována. Identifikací se tedy rozumí, převedení řízeného systému na jeho matematický model. Soustavu lze identifikovat dvěma způsoby:

- analyticky,
- experimentálně.

Při analytické identifikaci se matematický model systému získá sestavením rovnic vyjadřujících vztahy mezi vstupními, výstupními a stavovými signály prvků. Analytické metody vyžadují velké teoretické a praktické znalosti o identifikovaném systému. Získaný matematický model lze charakterizovat jako vnitřní popis zkoumaného systému. [3]

Při experimentální identifikaci se matematický model určí z experimentálně získaných údajů o měřeném systému. Tento matematický model lze charakterizovat jako vnější popis systému. Vnější popis vyjadřuje závislost mezi vektorem výstupního a vektorem vstupního signálu (přenos systému). [3]

Pro identifikování fyzikálního modelu kuličky na tyči byla použita analytická metoda.

### 7.1 Matematický popis modelu

Model kuličky na tyči byl na Katedře kybernetiky a biomedicínského inženýrství před několika lety řešen. Proto byl matematický popis modelu pro účely této diplomové práce ze vzniklých materiálů převzat. Materiály jsou uvedeny v literatuře [2]. V textu níže je pouze převzat základní obsah o postupu identifikace a její výsledek.

Základ matematického popisu modelu tvoří lagrangeovské pojetí mechaniky ve formě Lagrangeových rovnic II. druhu, umožňujících vytvoření pohybových rovnic soustavy hmotných bodů zavedením tzv. zobecněných souřadnic. Nejprve je nutno nalézt matematické relace pro kinetickou energii  $E_k$  a potenciální energii  $E_p$  soustavy kuličky na tyči. Ve výsledné kinetické energii soustavy uvažujeme čtyři druhy kinetické energie – dva druhy pro kuličku (translace, rotace) a dva druhy pro rameno, respektive tyč. [2]

Výsledná kinetická energie soustavy  $E_k$  je dána součtem kinetických energií kuličky  $E_{k1}$  a kinetických energií tyče bez kuličky a s kuličkou  $E_{k2}$ :

$$E_k = E_{k1} + E_{k2} = \frac{1}{2} \cdot \dot{x}^2(t) \cdot \left[ m_k + \frac{J_k}{R_k^2} \right] + \frac{1}{2} \cdot \dot{\varphi}^2(t) \cdot [J_t + m_k \cdot x^2(t)] \quad (3)$$



kde	$x(t) \equiv l_b(t)$	okamžitá poloha kuličky na tyči	[m]
	$\dot{x}(t) \equiv \dot{l}_b(t)$	okamžitá rychlost kuličky na tyči	$[m \cdot s^{-1}]$
	$\varphi(t)$	okamžitý úhel (fáze) sklonu tyče	[rad]
	$\dot{\varphi}(t)$	okamžitý úhlový kmitočet tyče	$[rad \cdot s^{-1}]$
	$J_t$	moment setrvačnosti tyče (ramene)	$[kg \cdot m^2]$

Výsledná potenciální energie soustavy  $E_p$  je dána relací:

$$E_p = F_p \cdot x(t) = m_k \cdot g \cdot x(t) \cdot \sin(\varphi) \quad (4)$$

V dalším kroku je nutno určit práci nekonzervativních sil  $W(t) \equiv W$ , tj. Lagrangeova rovnice II. druhu (v Eulerově-Lagrangeově tvaru) nebude homogenní čili rovna nule a bude reprezentována tzv. zobecněnou potenciálovou funkcí  $U[\dot{x}(t), x(t), t]$ , respektive  $U[\dot{\varphi}(t), \varphi(t), t]$ , přičemž zobecněnými souřadnicemi jsou okamžitá poloha kuličky  $x(t)$  a okamžitý úhel sklonu tyče  $\varphi(t)$ . [2]

Lagrangeova rovnice II. druhu pro nekonzervativní systém:

$$\frac{d}{dt} \left\{ \frac{\partial}{\partial \dot{q}_i(t)} L(t) \right\} - \frac{\partial}{\partial q_i(t)} \{L(t)\} = \frac{\partial}{\partial q_i} \{W(t)\} \quad (5)$$

kde	$L(t) \equiv L = E_k - E_p$	lagrangián	[J]
	$q_{i=1}(t) = x(t)$	okamžitá poloha kuličky	[m]
	$\dot{q}_{i=1}(t) = \dot{x}(t)$	okamžitá rychlost kuličky na tyči	$[m \cdot s^{-1}]$
	$q_{i=2}(t) = \varphi(t)$	okamžitý úhel (fáze) sklonu tyče	[rad]
	$\dot{q}_{i=2}(t) = \dot{\varphi}(t)$	okamžitý úhlový kmitočet tyče	$[rad \cdot s^{-1}]$

Po derivacích a dosazení do (4) dostáváme následující relace pro okamžitou polohu kuličky:

$$\frac{\partial}{\partial x(t)} \{W(t)\} = \left[ m_k + \frac{J_k}{R_k^2} \right] \cdot \ddot{x}(t) - m_k \cdot x(t) \cdot \dot{\varphi}^2(t) + m_k \cdot g \cdot \sin(\varphi) \quad (6)$$

a pro okamžitý úhel sklonu tyče:

$$\frac{\partial}{\partial \varphi(t)} \{W(t)\} = [J_t + m_k \cdot x^2(t)] \cdot \ddot{\varphi}(t) + m_k \cdot g \cdot x(t) \cdot \cos(\varphi) \quad (7)$$

Vliv náklonu na pohyb kuličky:

$$-b \cdot \dot{x}(t) - \frac{\xi}{R_k} \cdot F_n \cdot \text{sign}[\dot{x}(t)] = \left[ m_k + \frac{J_k}{R_k^2} \right] \cdot \ddot{x}(t) - m_k \cdot x(t) \cdot \dot{\varphi}^2(t) + m_k \cdot g \cdot \sin(\varphi) \quad (8)$$

Pro odvození přenosové funkce této soustavy (vztah mezi náklonem tyče a pozicí kuličky) budeme dále uvažovat relaci (7), kterou lze zredukovat zanedbáním ramena valivého odporu a odstředivé síly kuličky a zlinearizovat okolo následujícího ekvilibria: [2]

$$\varphi(t) = \dot{\varphi}(t) = x(t) = \dot{x}(t) = 0 \quad (9)$$

Při linearizaci dále uvažujeme:

$$\sin(\varphi) \cong \varphi \quad \varphi \in \langle 0, \frac{\pi}{36} \rangle \text{ [rad]} \quad (10)$$

Při uvažování relací (8) a (9) lze relaci (7) přepsat do tvaru homogenní OLDR 2. řádu:

$$\left[ m_k + \frac{J_k}{R_k^2} \right] \cdot \ddot{x}(t) + b \cdot \dot{x}(t) + m_k \cdot g \cdot \varphi = 0 \quad (11)$$

Přenosová funkce (vnější popis) v Laplaceově transformaci při nulových počátečních podmínkách a volbou nulového odporu prostředí je následně dána relací:

$$\begin{aligned} \hat{G}_s(s) &= \frac{\hat{L}\{x(t)\}}{\hat{L}\{\varphi(t)\}} = \frac{\hat{X}(s)}{\hat{\Phi}(s)} = \frac{-m_k \cdot g}{\left[ m_k + \frac{J_k}{R_k^2} \right] \cdot s^2 + b \cdot s} = \frac{-F_g}{\left[ m_k + \frac{J_k}{R_k^2} \right] \cdot s^2 + b \cdot s} = \\ &= \frac{b_0}{a_2 \cdot s^2 + a_1 \cdot s} \Rightarrow \frac{-0,550}{0,077 \cdot s^2} = \frac{-7,1428}{s^2} \end{aligned} \quad (12)$$

Stavový popis (vnitřní popis) lze získat substitucí lineární diferenciální rovnice 2. řádu v relaci (10) na soustavu dvou lineárních diferenciálních rovnic 1. řádu:

$$\dot{x}_1(t) = \frac{1}{\left[ m_k + \frac{J_k}{R_k^2} \right]} \cdot [-b \cdot x_1(t) - m_k \cdot g \cdot u(t)] \quad (13)$$

$$\dot{x}_2(t) = x_1(t) \quad (14)$$

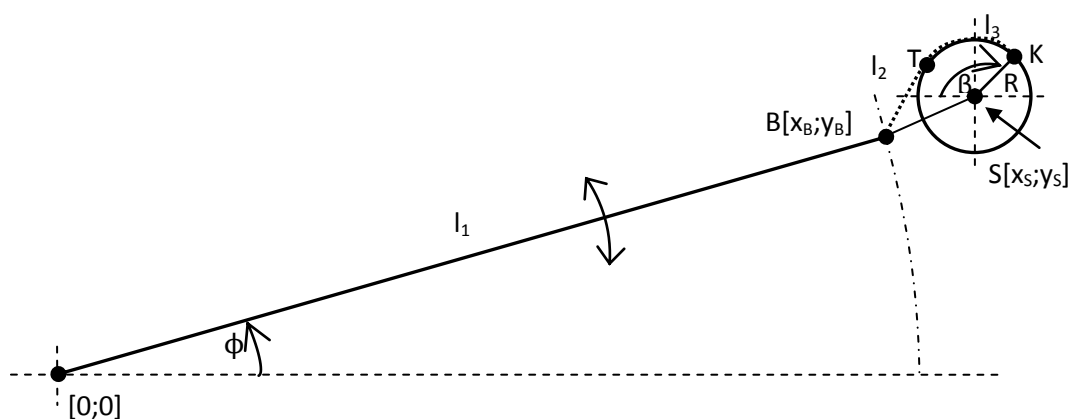
při substituci:

$$x_1(t) = \dot{x}(t); x_2(t) = x(t); u(t) = \varphi(t) \quad (15)$$

Tímto jsme získali astatický systém 2. řádu. Vzhledem k původní rovnici (7) lze očekávat nelineární chování kuličky na okrajích nakláněného ramene, kde výrazněji působí zanedbaná odstředivá síla. Další významnou nelinearitou je samozřejmě konečná délka ramene a v praxi se projeví i nerovnosti vodících kolejnic (odporových drátů tvořících odporovou dráhu). [2]

## 7.2 Přepočet úhlu ramene na úhel servomotoru

Podkapitola se věnuje přepočtu úhlu zvednutí, resp. úhlu poklesu ramena na úhlu natočení servomotoru, na kterém je mechanicky připevněna odporová dráha s pohybující se kuličkou. Správným poměrem těchto dvou úhlů se získá i správná regulace pohybu natočení servomotoru. Následující obrázek popisuje, jak se bude chovat fyzikální model při regulaci (Obr. 7.2a).



Obr. 7.2a: Závislost úhlu ramene na natočení servomotoru

Hledáme:

$$\phi = f(\beta) \rightarrow \beta = f(\phi) \quad (16)$$

Souřadnice konce ramene:

$$B [l_1 \cdot \cos\phi ; l_2 \cdot \sin\phi] \quad (17)$$

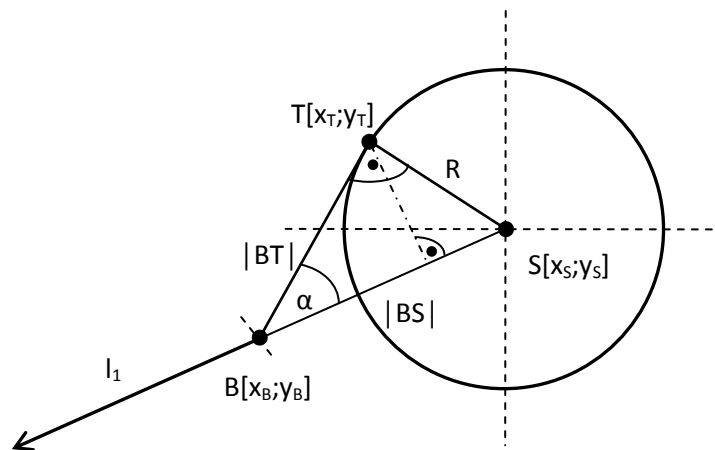
Souřadnice natočení servomotoru:

$$K [X_S + R \cdot \cos\beta ; Y_S + R \cdot \sin\beta] \quad (18)$$

Výpočet délky provázku od T do K:

$$l_3 = R \cdot \beta \quad (19)$$

Zvětšení detailu kolečka, které je součástí hřídele servomotoru, je na Obr. 7.2b.



Obr. 7.2b: Detail kolečka servomotoru

Výpočet vzdáleností konce ramene od středu kolečka:

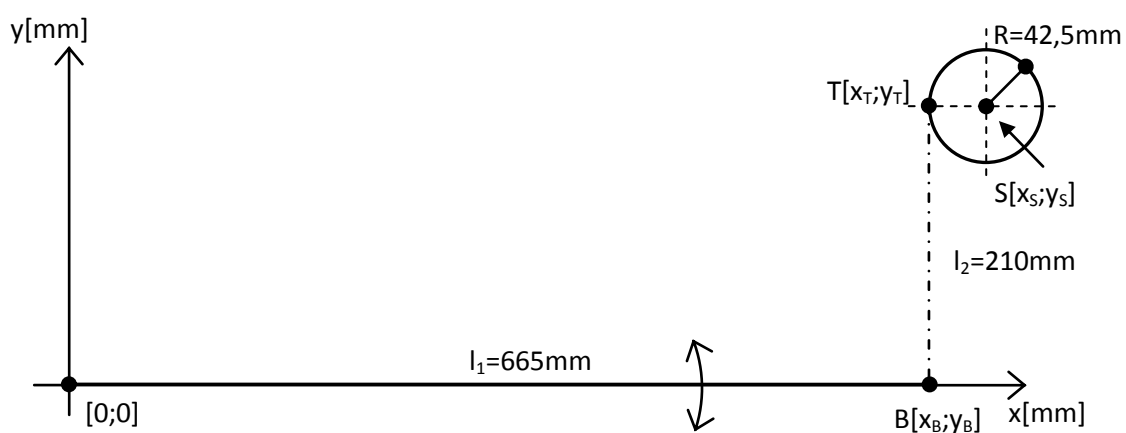
$$|BS| = \sqrt{(x_S - x_B)^2 + (y_S - y_B)^2} \quad (20)$$

$$|BT| = \sqrt{|BS|^2 - R^2} \rightarrow \sqrt{(x_S - x_B)^2 + (y_S - y_B)^2 - R^2} \quad (21)$$

$$\sqrt{(x_T - x_B)^2 + (y_T - y_B)^2} = \sqrt{(x_S - x_B)^2 + (y_S - y_B)^2 - R^2} \quad (22)$$

$$(x_T - x_B)^2 + (y_T - y_B)^2 = R^2 \quad (23)$$

Fyzikální model v počátečním stavu, kdy je rameno vodorovně vůči povrchu země, je zakreslen na Obr. 7.2c.



Obr. 7.2c: Model v počátečním stavu

## 7.2.1 Přepočet úhlů v MATLABu

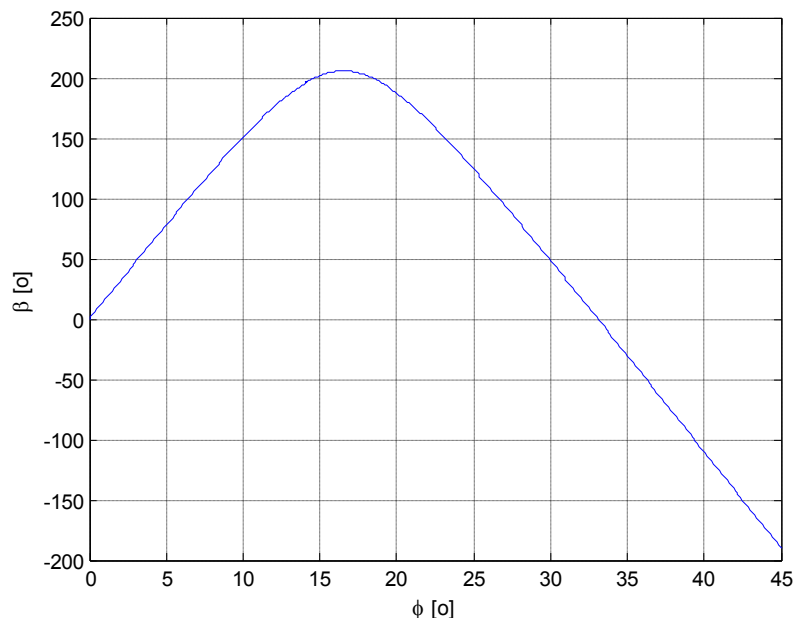
Přepočet úhlu ramene  $\phi$ , na kterém je umístěna odporová dráha, na úhel natočení servomotoru  $\beta$ , byl proveden v programu MATLAB. Níže je uveden okomentovaný zdrojový kód s výsledkem přepočtu v grafu.

```
syms xt yt xs ys xb yb R

l1=665; % délka tyče odporové dráhy
xs_value=707; % souřadnice délky středu kolečka servomotoru
ys_value=210; % souřadnice délky středu kolečka servomotoru
R_value=42; % poloměr kolečka servomotoru
delka_lanko=210; % délka provázku od tečny serva k tyči v 0 stupních

% úhel fi
fi_stupne=0:0.1:45;
fi=fi_stupne*2*pi/360;
xb_value=l1*cos(fi);
yb_value=l1*sin(fi);

% úhel beta
BT=sqrt((xs_value-xb_value).^2+(ys_value-yb_value).^2-R_value^2);
l4=delka_lanko-BT;
beta=l4/R_value;
beta_stupne=beta*360/2/pi;
% vykreslení grafu
plot(fi_stupne,beta_stupne),grid
xlabel('\phi [o]')
ylabel('\beta [o]')
```



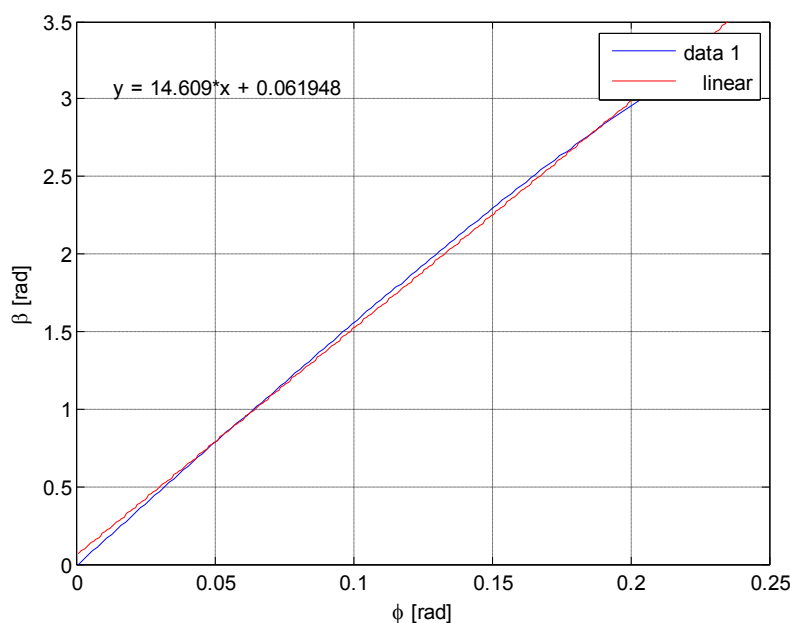
Graf 7.2.1: Přepočet úhlu ramene na úhel servomotoru

Z grafu 7.2.1 je vidět, že pro zvednutí ramene odporové dráhy o úhel  $10^\circ$  je potřeba, aby se servomotor natočil o úhel  $150^\circ$  a maximální natočení servomotoru odpovídá zvednutí odporové dráhy z nulové pozice do úhlu  $12,4^\circ$ .

## 7.3 Linearizace

Všechny dynamické procesy jsou popsány nelineárními diferenciálními rovnicemi. Protože, ale matematika nemá přesné univerzální matematické metody pro řešení těchto rovnic, nabízí se možnost popsat nelineární dynamický systém lineární diferenciální rovnicí tak, aby byl popsán za jistých okolností s přijatelnou přesností. Toto je základní myšlenka linearizace.

Linearizace v pracovním bodě vychází z matematicko-fyzikálního modelu kuličky na tyči. Z předchozí podkapitoly byl získán graf zobrazující vztah mezi úhlem natočení ramene a úhlem natočení servomotoru. Nyní se užitečná část grafu vykreslí v radiánech a proloží lineární přímkou, jak je vidět níže.

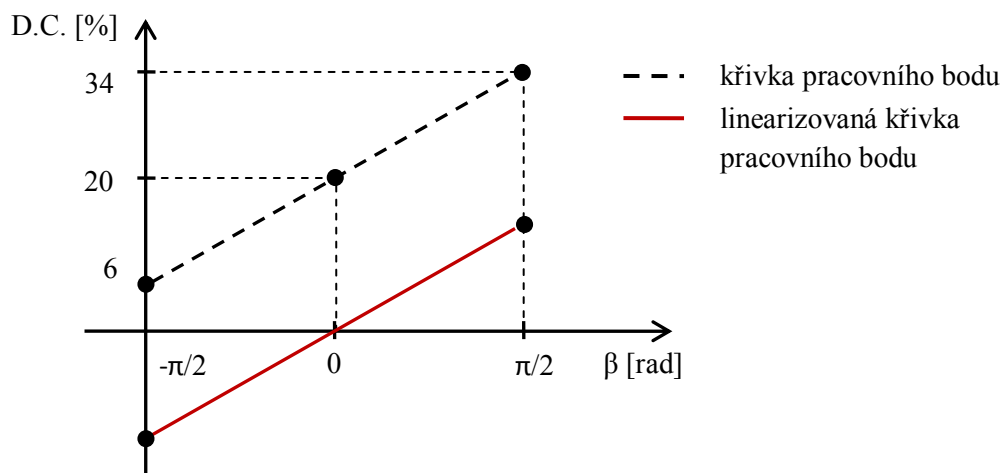


**Graf 7.3.1: Lineární vykreslení závislosti**

K získání lineárního průběhu byl v MATLABu použit nástroj Basic Fitting. Získaný vztah je vidět v grafu. Tento vztah dává zesílení, které slouží k lineárnímu přepočtu úhlu zdvihu ramene na úhel natočení servomotoru.

Další linearizace se provádí u přepočtu úhlu natočení servomotoru na velikost střídý PWM.

Vychází se z následujícího obrázku, kde čárkovaná čára představuje původní křivku pracovního bodu a červená čára představuje posunutí pracovního bodu do počátku.

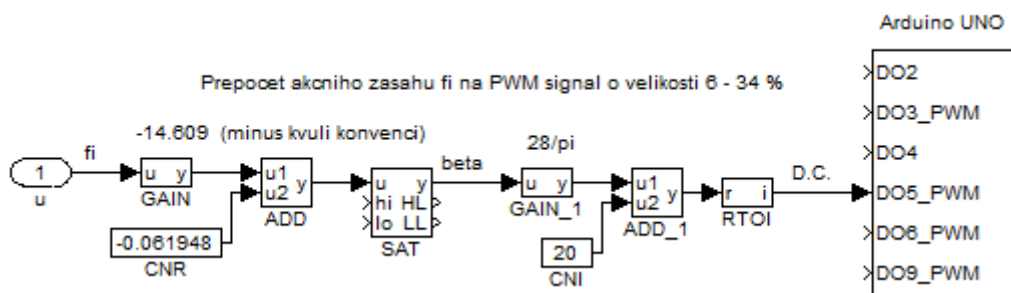


Obr. 7.3.1: linearizace v pracovním bodě

Vzniklý linearizovaný průběh odpovídá následujícímu vztahu:

$$D.C. = \frac{28}{\pi} * \beta + 20 \quad (24)$$

Výpočty linearizovaného pracovního bodu byly modifikovány do prostředí RexDraw, kde slouží k praktickému využití (Obr. 7.3.2).

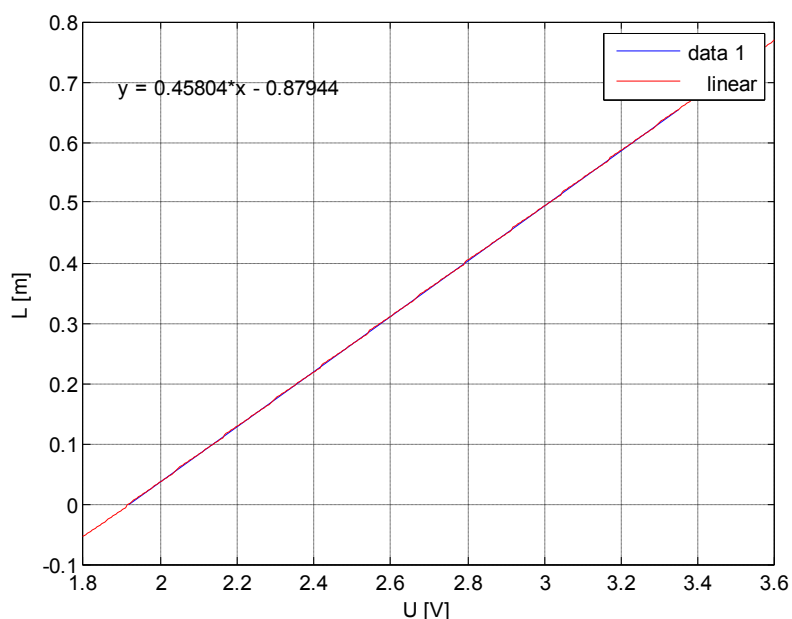


Obr. 7.3.2: Propojení bloků s linearizovanými vztahy v RexDraw

Na Obr. 7.3.2 představuje veličina  $u$  akční zásah regulátoru o velikosti náklonu ramene  $\phi$  [rad]. Za bloky pro přepočítání z  $\phi$  do  $\beta$  následuje saturační blok, který slouží k ořezání hodnot na

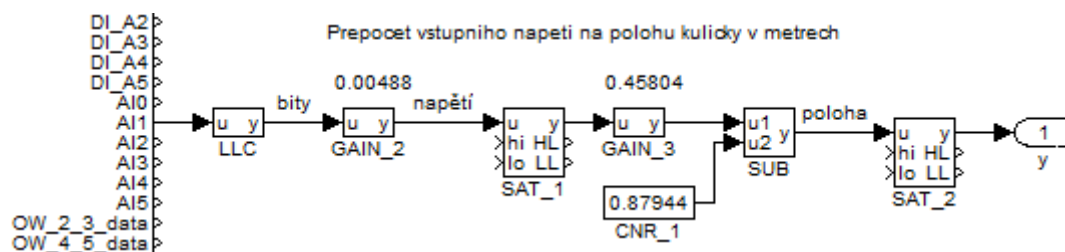
velikost  $\pm 1,5$ . Za saturačním blokem jsou propojeny bloky podle vztahu (24) a blok *RTOI* slouží k zaokrouhlování na nejbližší celé číslo. Za tímto blokem získáme hodnotu střidy PWM signálu, která je generována na výstupu Arduina.

Měření polohy kuličky se provádí přivedením analogového signálu na analogový vstup Arduina. Minimální hodnota napětí změřená na dráze je  $1,92V$ , ta odpovídá počáteční vzdálenosti tedy hodnotě  $0m$ . Maximální hodnota napětí na dráze je  $3,35V$ , což odpovídá poloze kuličky ve vzdálenosti  $0,665m$ . Graf 7.3.2 zobrazuje výpočet linearizace křivky.



**Graf 7.3.2: Lineární vykreslení závislosti**

Zapojení bloků za účelem získání hodnoty polohy v metrech je vidět na následujícím obrázku.



**Obr. 7.3.3: Převod napětí na polohu v metrech**



Na analogový vstup *A11* je přivedeno napětí z elektroniky. Toto napětí Arduino převádí na hodnotu 0-1023 bitů. Kdy 0 odpovídá napětí  $0V$  a 1023 odpovídá napětí  $5V$ . Než se hodnota bitů dostane do bloku *GAIN\_2*, kde je přepočítána opět na napětí, projde přes filtr druhého řádu LLC. Saturační blok *SAT\_1* slouží pro nastavení mezí napětí. Následují bloky pro výpočet linearizace a saturační blok pro nastavení mezí polohy 0 až  $0,665m$ .

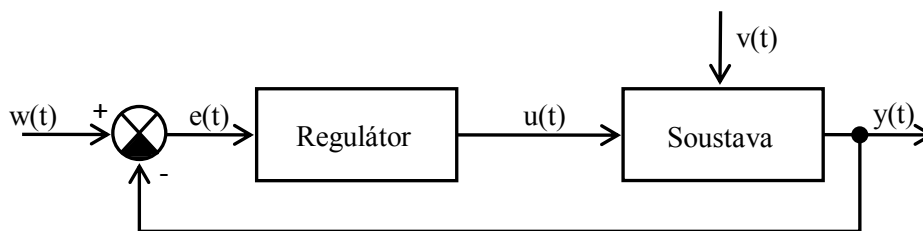
## 8 Návrh regulátoru

Diplomová práce se věnuje návrhu regulátorů pomocí tzv. moderních a tzv. postmoderních metod řízení. V textu níže jsou také uvedeny vztahy klasické metody regulace, jako jsou používané modifikace PID regulátoru. PID regulace je použita pro simulaci regulačního obvodu. V kapitole je popsán vznik moderní teorie řízení, jeho výhody oproti klasické teorii a metody návrhu regulátorů. K regulaci polohy kuličky na tyči se používá LQG řízení.

### 8.1 Klasická teorie řízení

Metoda klasické teorie řízení je v diplomové práci použita pro vizualizaci simulovaného procesu v prostředí MATLAB/Simulink, REX a v internetovém prohlížeči.

Regulací se nazývá řízení se zpětnou vazbou. Regulace má za úkol nastavení technických veličin (teplota, tlak, otáčky, vzdálenost atd.) na požadovanou hodnotu a na této hodnotě je udržet i při působení vnějších nebo vnitřních poruch, které mohou vznikat v různých místech regulačního obvodu (Obr. 8.1).



**Obr. 8.1: Spojitý zpětnovazební regulační obvod**

Základem regulátorů se zápornou zpětnou vazbou je prostřednictvím akční veličiny  $u$  působit na regulovanou soustavu tak, aby se regulovaná veličina  $y$  udržovala na předepsané hodnotě a regulační odchylka  $e$  byla nulová nebo co nejmenší. Regulátor musí být zapojen tak, aby akční veličina zmenšovala regulační odchylku. [3]

V regulačních obvodech se hodnota regulované veličiny trvale měří a porovnává se žádanou hodnotou a výsledným rozdílem je regulační odchylka  $e$ :

$$e(t) = w(t) - y(t) \quad (25)[1]$$

Regulátory se používají z důvodu působení různých poruchových veličin na regulovanou soustavu a tím ji nepříznivě ovlivňují. V Obr. 8.1 je nepříznivé ovlivňování regulované soustavy poruchovou veličinou označeno písmenem  $v$ .

### 8.1.1 Návrh PID regulátoru

Regulátor je obecně zařízení, které provádí samotnou regulaci. Působí akční veličinou  $u$  na regulovanou soustavu a udržuje regulační odchylku  $e$  na nulové nebo co nejnížší hodnotě. Regulátor může regulační odchylku zesilovat, integrovat a derivovat. V této závislosti mohou být navrhovány různé typy regulátorů (P, I, PI, PD, PID). Dále bude využit pouze PID regulátor.

Přenos ideálního PID regulátoru:

$$G_R(s) = P + \frac{I}{s} + Ds \rightarrow Kp * \left(1 + \frac{1}{T_i s} + T_d s\right) \quad (26)[1]$$

Regulátor PID (proporcionálně-integračně-derivační) je v této podkapitole seřízen v časové oblasti a využit pro regulaci matematického modelu kuličky na tyči. K seřízení regulátoru se používá příkaz „*pidtune*“ v MATLABU. Zadá se příslušný přenos soustavy  $G_s$  a příkaz pro výpočet PID regulátoru viz. níže okomentovaný kód.

```
Gs = tf(7.1428,[1 0 0]) % Přenos soustavy
Gr = pidtune(Gs,'PID') % Výpočet konstant regulátoru
```

MATLAB vypočítá hodnoty konstant ve tvaru:

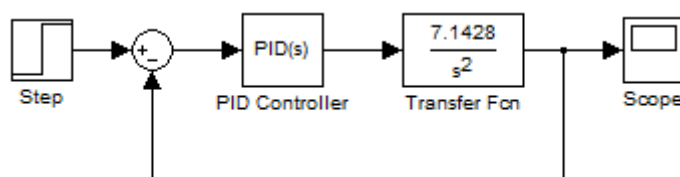
$$PID = K_p + K_i * \frac{1}{s} + K_d s \rightarrow P + \frac{I}{s} + Ds \quad (27)$$

$$K_p = 0,07$$

$$K_i = 0,0011876 \rightarrow T_i = 58,9226$$

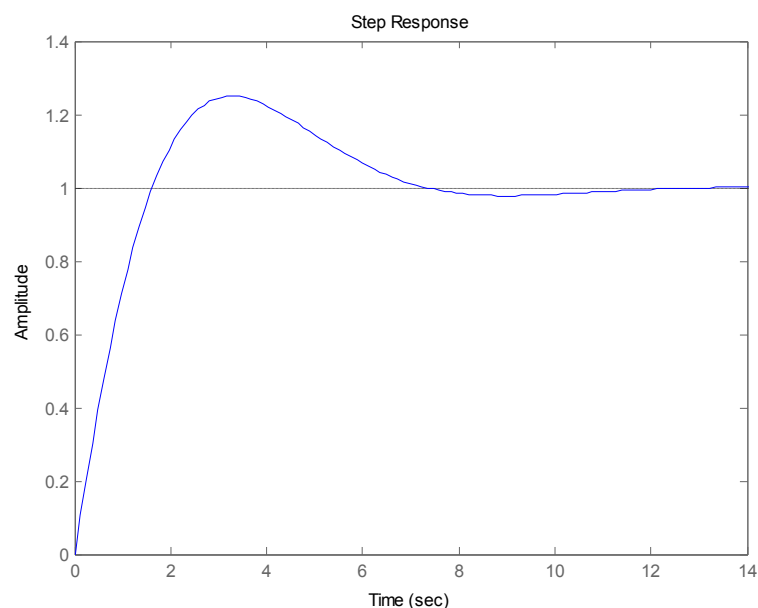
$$K_d = 0,12243 \rightarrow T_d = 1,7497$$

Vypočítané hodnoty konstant se následně implementují do bloku PID regulátoru v Simulinku a odzkouší se regulace (Obr. 8.1.1).



Obr. 8.1.1: Regulační obvod s PID regulátorem v MATLAB/Simulink

Vstupním signálem je jednotkový skok přivedený na vstup regulátoru z bloku *Step*. Výsledná odezva je zobrazena v následujícím grafu.



**Graf 8.1.1: Odezva systému na jednotkový skok**

## 8.2 Moderní teorie řízení

Pojem „moderní teorie řízení“ vznikl v 60. letech k odlišení přístupu založeném na stavovém popisu systémů od „klasické teorie“, vycházející téměř výhradně z popisu vnějšího. V současné době se oba přístupy (stavový a přenosový) silně prolínají. Charakteristickou vlastností moderní teorie řízení však zůstává využívání matematických modelů k popisu řízených procesů. Je-li přesně specifikován model, inženýrskou úlohu návrhu regulátoru lze formulovat jako úlohu optimalizační. Místo přímého nastavování konstant regulátoru tedy inženýr nastavuje parametry kritéria optimality a případná omezení. Tento přístup je výhodný v tom, že některé vlastnosti získaných řešení (např. stabilita při kvadraticky optimálním řízení) jsou zajištěny implicitně. Spoléhání na přesný model s sebou přináší i některé nedostatky moderní teorie řízení, které motivují i nové směry výzkumu tzv. robustních metod, které dovoluují zahrnout do formulace problému i neurčitost matematického modelu. [1]

### 8.2.1 Problematika optimálního lineárně kvadratického řízení

V moderní teorii řízení se úloha návrhu regulátoru formuluje jako optimalizační problém. Ladění regulátoru se pak provádí nastavováním parametrů optimalizačního kritéria. Tato problematika je vyvíjena od 60 let a nazývá se Lineárně kvadratické řízení (LQ).

V klasických metodách návrhu se požadovaných vlastností systému (stabilita, požadovaná šířka pásma, amplitudová a fázová bezpečnost) dosahovalo vhodným nastavením konstant regulátorů. Moderní teorie řízení splňují řadu těchto požadavků implicitně a v určitém smyslu vedou na tzv. optimální regulátor. Tento regulátor hledá minimum nebo maximum účelové funkce a říkáme, že účelovou funkci optimalizuje. Nejčastěji se jako účelová funkce používá kvadratické kritérium.

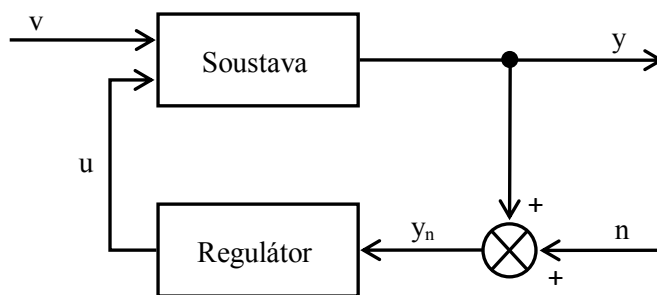
U regulované soustavy se předpokládá, že je lineárním systémem popsaným vnitřním popisem stavů systému a objektem řízení je kvadratický funkcionál stavů systému a řídicích vstupů - akčních veličin. Hlavním úkolem tohoto řízení je minimalizovat kvadratický funkcionál s ohledem na akční veličiny - řízení. LQ řízení má za úkol získat stabilní optimální systém s přiměřeně dobrou odezvou. U těchto systému se předpokládá a požaduje, aby díky regulátoru byly systémy v rovnováze, nebo nastaveny do daného bodu navzdory rušení. Proto je hlavním cílem minimalizovat působení rušení na daný systém.

Výhodou formulace problémů LQ řízení je výsledek získaný z lineárních zákonů regulace, které jsou jednoduché pro implementaci a analýzu. Dalšími možnými aplikacemi LQ řízení mohou být požadavky na minimální čas řízení, minimální spotřebu energie/paliva (kde řešení vyžaduje nelineární typ řízení zapnuto/vypnuto, které je jednoduché k implementaci užitím reléových spínačů nebo přepínačů, ale obtížnější k analýze nelinearit), a ITAE kritérium. [1]

### 8.2.2 LQG řízení

LQG řízení je zkratka označující optimální lineárně kvadratické Gaussovské řízení. Představuje moderní techniku návrhu optimálních dynamických regulátorů ve stavovém prostoru a umožňuje vyvážit poměr výkonu regulace a řízení při uvažování procesního a měřicího šumu.

Při návrhu LQG řízení musí být určen stavový model systému. Uspořádání zpětnovazebního obvodu pro LQG řízení je na Obr. 8.2.2.



**Obr. 8.2.2: Uspořádání zpětnovazebního obvodu pro LQG řízení**

LQG regulátor je principiálně tvořen optimálním zpětnovazebním tvarem odpovídající LQ regulaci a Kalmanovým odhadem stavu. Tyto dvě komponenty mohou být navrhovány nezávisle.

Soustava je vystavena procesnímu šumu  $v$  a řízení  $u$ . Regulátor provádí regulaci na základě zašuměného měření dle literatury [2]:

$$y_{n(t)}(t) = y(t) + n(t) \quad (28)$$

kde  $n(t)$  náhodný proces (bílý šum)

Časově invariantní soustavu lze následně popsat ve tvaru stavové rovnice a měřicí (výstupní) rovnice následovně:

$$\begin{aligned} \text{stavová rovnice: } \dot{x}(t) &= \hat{A} \cdot x(t) + \hat{B} \cdot u(t) + \hat{G} \cdot v(t) \\ \text{výstupní rovnice: } y_{n(t)}(t) &= \hat{C} \cdot x(t) + \hat{D} \cdot u(t) + \hat{H} \cdot v(t) + n(t) \end{aligned} \quad (29)$$

kde  $\hat{A}$  matice vnitřních vazeb soustavy (systémová matice)  
 $\hat{B}$  matice vstupů a stavů (vstupní matice)  
 $\hat{C}$  matice stavů a výstupů (výstupní matice)  
 $\hat{D}$  matice přímých vazeb vstupů a výstupů  
 $\hat{G}$  rozšiřující matice procesní poruchy ve stavové rovnici  
 $\hat{H}$  rozšiřující matice procesní poruchy v měřicí rovnici  
 $x(t)$  vektor stavů  
 $u(t)$  vektor řízení

Chování LQG regulace se hodnotí lineárně-kvadratickým kritériem ve tvaru funkcionálu:

$$J[x(t), u(t), t] = \int_0^T \{x^{tr}(t) \cdot \hat{Q} \cdot x(t) + 2 \cdot x^{tr}(t) \cdot \hat{N} \cdot u(t) + u^{tr}(t) \cdot \hat{R} \cdot u(t)\} \cdot dt \quad (30)$$

kde  $\hat{Q}, \hat{N}, \hat{R}$  specifikované váhové matice přerozdělení vlivů stavů nebo řízení  
 $\hat{Q}_n = E\{v(t) \cdot v^{tr}(t)\}; \hat{N}_n = E\{v(t) \cdot n^{tr}(t)\}; \hat{R}_n = E\{n(t) \cdot n^{tr}(t)\}$

Řešením optimalizačního kritéria (29) je následující relace minimalizující tento funkcionál, tedy:

$$u(t) = -\hat{K} \cdot x(t) \quad (31)$$

kde  $\hat{K}$  zes. min. matice odvozené z řešení Riccatiho rovnice (optimální LQ zesílení)

Aby bylo možno relaci (31) implementovat, je třeba znát kompletní informaci o stavech. Pokud však tuto informaci nemáme k dispozici, můžeme ji stavovým estimátorem (Kalmanův filtr) odhadnout z chování původní soustavy:

$$\hat{\hat{x}}(t) = \hat{A} \cdot \hat{x}(t) + \hat{B} \cdot u(t) + \hat{L} \cdot \{y_{n(t)}(t) - \hat{C} \cdot \hat{x}(t) - \hat{D} \cdot u(t)\} \quad (32)$$

kde  $\hat{x}(t)$  odhad vektoru stavů  
 $\hat{L}$  matice pozorovatele

Kalmanův filtr tedy zajišťuje optimální odhad vektoru stavů při zarušení gaussovským (bílým) šumem (procesní porucha, šum měření), neboli minimalizuje chybovou kovarianci mezi vektorem stavů a odhadnutým vektorem stavů:

$$\lim_{t \rightarrow +\infty} E\{[x(t) - \hat{x}(t)] \cdot [x(t) - \hat{x}(t)]^{tr}\} = 0 \quad (33)$$

Stavové rovnice LQG regulátoru lze pomocí relací (30) a (31) získat v následujícím tvaru:

$$\hat{\hat{x}}(t) = \{\hat{A} - \hat{L} \cdot \hat{C} + [-\hat{B} + \hat{L} \cdot \hat{D}] \cdot \hat{K}\} \cdot \hat{x}(t) + \hat{L} \cdot y_{n(t)}(t) \quad (34)$$

$$u(t) = -\hat{K} \cdot \hat{x}(t) \quad (35)$$

### 8.2.3 Návrh LQG regulátoru

V matematickém popisu LQG řízení výše je uveden analytický postup řešení, které může být s výhodou implementováno do programu MATLAB v podobě skriptu v m-souboru. Definujeme vstupní parametry úlohy a algoritmus návrhu: [2]

- Vstupními parametry úlohy jsou:

stavové matice kuličky na tyči – Frobeniův kanonický tvar (přímé programování):

$$\hat{A} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}; \hat{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \hat{C} = \begin{bmatrix} 0 & \frac{-m_k \cdot g}{m_k + \frac{J_k}{R_k^2}} \end{bmatrix}; \hat{D} = [0] \quad (36)$$

rozšiřující váhové matice:

$$\hat{G} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \hat{H} = [0] \quad (37)$$

kovariance procesního šumu:

$$\hat{Q}_n = E\{v(t) \cdot v^{tr}(t)\} = [0,02] \quad (38)$$

kovariance šumu měření:

$$\hat{R}_n = E\{n(t) \cdot n^{tr}(t)\} = [0,02] \quad (39)$$

vzájemná kovariance procesního šumu a šumu měření:

$$\hat{N}_n = E\{v(t) \cdot n^{tr}(t)\} = [0] \quad (40)$$

- Algoritmus návrhu. S využitím funkcí `kalman`, `lqr`, `lqgreg` a `lqr_kompence` lze definovat Kalmanův filtr, výpočet optimálního LQ zesílení  $\hat{K}$ , sestavení LQG regulátoru a výpočet hodnoty kladného skaláru  $\gamma_0$ : [2]

Kalmanův filtr:

```
[Kalmf, L, P] = kalman(Gs_ss, QN, RN);
```

výpočet optimálního LQ zesílení:

```
[K, S, E]=lqr(A, B, Q, R);
```

sestavení LQG regulátoru:

```
Kreg = lqgreg(Kalmf, K);
```

hodnota kladného skaláru:

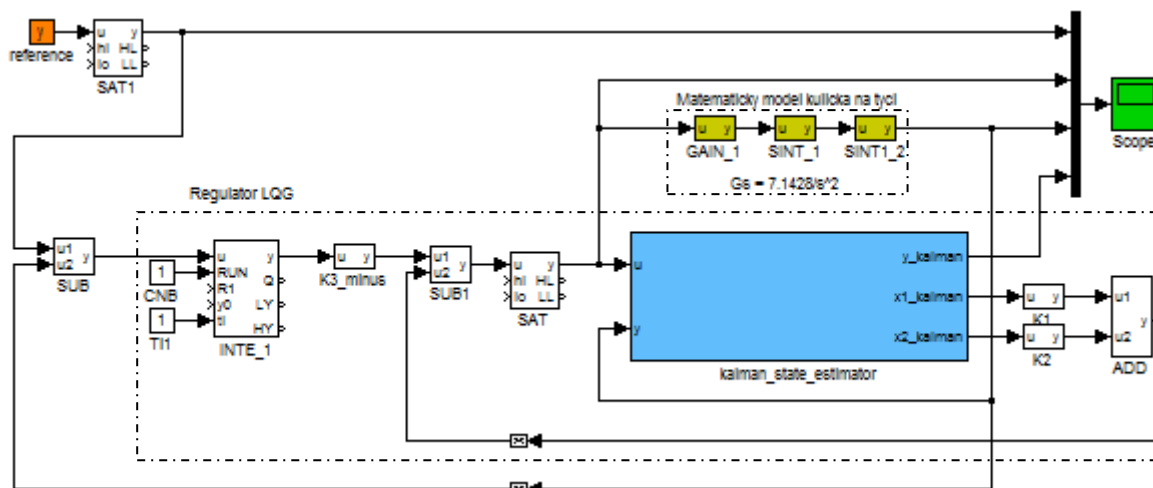
```
Ng = lqr_kompence(A, B, C, D, K);
```

V příloze II. jsou dodány soubory typu `.m`, které kompletně řeší daný problém návrhu regulátoru LQG. Tyto soubory byly převzaty z literatury [2].

Simulace navrženého LQG regulátoru s vypočítanými konstantami v prostředí Simulink je zobrazena na Obr. 8.2.3. K regulátoru s Kalmanovým odhadcem stavů je připojen matematický model regulované soustavy a všechny signály regulačního obvodu jsou přivedeny na blok *Scope*. K namodelování byly použity bloky z knihovny REXLib, které jsou se Simulinkem kompatibilní. Z bloku *reference* se na vstup regulátoru přivádí žádaná hodnota v metrech, která je v saturačním bloku satureována na hodnotu 0 až 0,665m. Tato délka odpovídá skutečné délce ramene odporové dráhy.

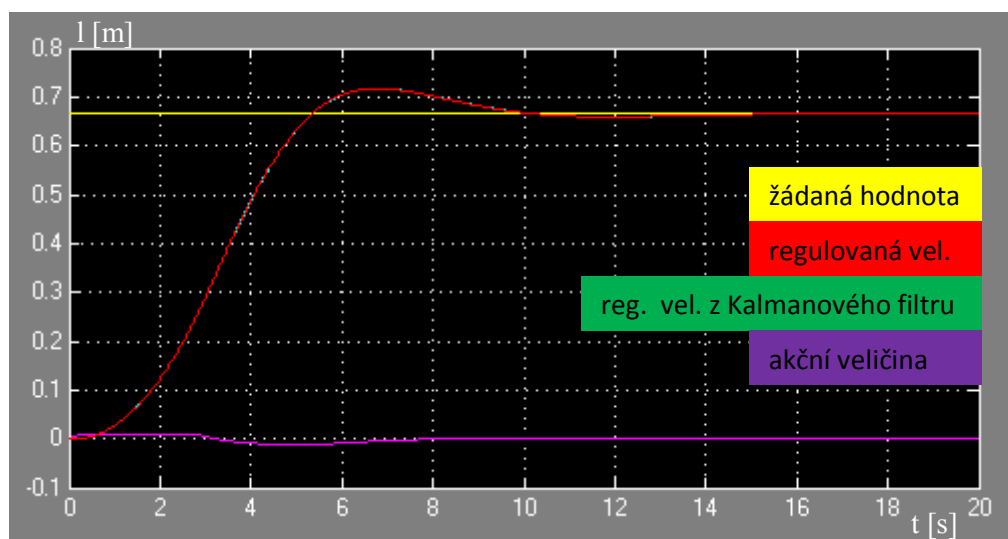
Před zapnutím simulace je důležité nastavit konfigurační parametry spouštění úlohy. V REXu by program běžel s periodou nastavenou v hlavním souboru (20ms). To je nezbytné nastavit i u simulace v Simulinku. Nastavení periody cyklu se provede vybráním záložky *Simulation/Configuration Parameters/Solver* a zvolí se *Type: Fixed-step, Solver: discrete (no continuous states)*, *Fixed-step size: 0,02*. Hodnota 0,02 odpovídá periodě spouštění úlohy simulace v řídicím systému REX.





Obr. 8.2.3: Namodelovaný regulační obvod s LQG regulátorem v Simulinku

Obr. 8.2.3 je pro přehlednost součástí přílohy II, kde je zobrazen ve větším měřítku. Průběh regulačního děje s nastavením maximální hodnoty reference je vidět v grafu 8.2.3.



Graf 8.2.3: Průběh regulace s LQG regulátorem

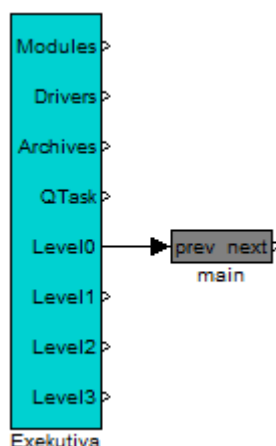
V čase 0s je do regulačního obvodu přivedena žádaná hodnota o velikosti 0,665m. V bloku *SUB* je vypočítán rozdíl mezi žádanou hodnotou a regulovanou veličinou. Získaná regulační odchylka přichází na vstup regulátoru a je vykonán příslušný akční zásah. Zelená křivka představující regulovanou veličinu vystupující z Kalmanova filtru. V grafu není dobře vidět, protože splývá s regulovanou veličinou, která je zobrazena červeně.

## 9 Realizace regulátoru v řídicím systému REX

V řídicím systému REX byly navrženy regulační obvody s regulátorem LQG a regulátorem PID. Oba regulátory regulují matematický model kuličky na tyči a fyzikální model kuličky na tyči (reálná soustava) je regulován pouze LQG regulátorem. V textu je popsáno vytvoření projektu regulačního obvodu v řídicím systému REX verze 2.08.

Konfigurace programu regulačního obvodu probíhá v prostředí RexDraw. Projekt je složen ze dvou souborů *.mdl*. První soubor je hlavní soubor projektu, ve kterém se konfiguruje cílové zařízení, ovladače, priority, moduly a další. Definování prvního souboru, jako hlavního souboru projektu, se provede v menu volbou *File/Set as main*. Po tomto kroku se změní jméno okna na jméno ve formátu *nazev - RexDraw [nazev.rex]*. [3]

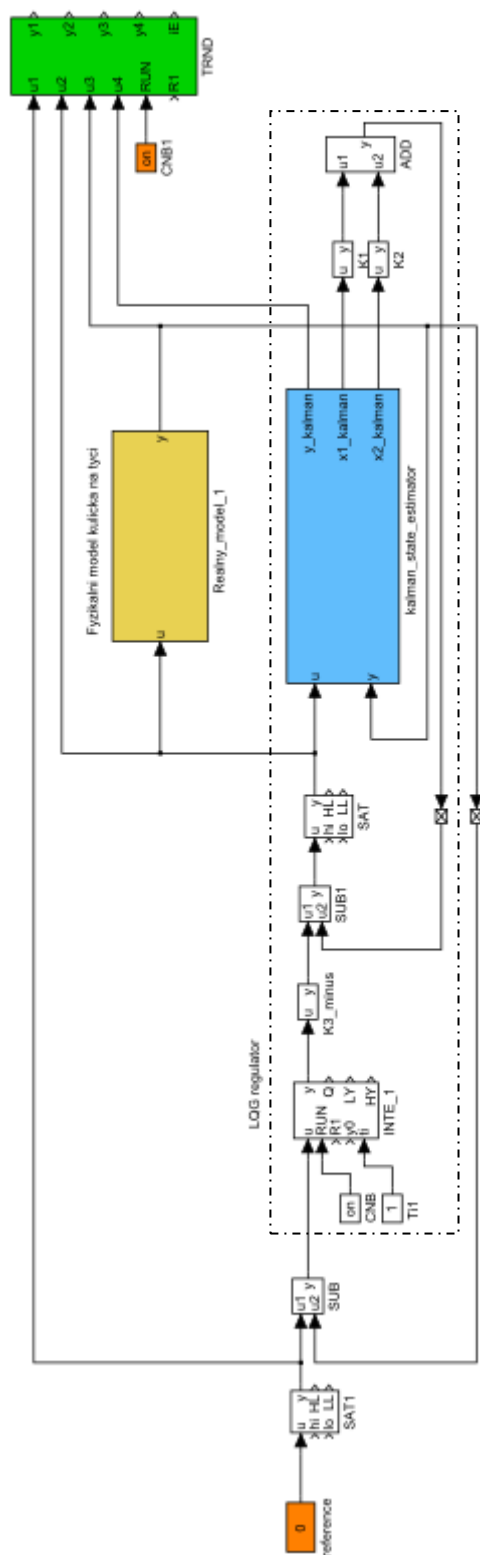
Hlavní soubor musí mít nadefinované cílové zařízení (*target*), ve kterém bude program běžet. V případě diplomové práce běží program na řídicím počítači ALIX, který pracuje na OS Linux. Perioda spouštění programu (tzv. tick) byla nastavena na hodnotu 0,016384s. Tato hodnota odpovídá frekvenci *PWM* výstupu Arduina.



Obr. 9a: Hlavní soubor projektu

Druhý soubor projektu obsahuje samotný algoritmus regulačního obvodu. Nutnou součástí před spuštěním programu je jeho přeložení. Příklad je spuštěn z menu *Compiler/Compile*. O výsledku překladače je uživatel informován v okně *Compiler window*. Při bezchybném překladači se vytvoří soubor *nazev.rex*. [3]

Po překladači lze nový binární konfigurační soubor nahrát programem RexView do cílového zařízení – ALIXU. V okamžiku nahrávání musí na řídicím počítači běžet jádro řídicího systému REX (RexCore). Obrázek regulačního obvodu s LQG regulátorem je vidět níže.



Obr. 9b: Regulační obvod v řídicím systému REX

Blok s názvem *Realny\_model\_1* je blok subsystému, ve kterém jsou vloženy bloky pro linearizaci v pracovním bodě a blok vývojové platformy Arduino. Blok *kalman\_state\_estimator* má nastavené parametry vah pomocí výpočtů v MATLABu. Bloky *K1*, *K2*, *K3\_minus* jsou MATLABem vypočítané stavové konstanty LQG regulátoru.

Programem RexView byl konfigurační soubor úlohy nahrán do cílového zařízení. Tento program obsahuje i automatický trend, ve kterém uživatel pozoruje průběhy veličin regulačního obvodu v čase. Bohužel se trendu nedá úplně dobře nastavit časová osa, proto jsou průběhy regulace s fyzikálním modelem kuličky na tyči zobrazeny v kapitole Vizualizace.

Regulační obvody s PID a LQG regulátorem, které regulují matematický model kuličky na tyči, byly opět vytvořeny v řídicím systému REX a slouží pro účely vizualizace simulace.

# 10 Vizualizace

Součástí diplomové práce je vizualizace regulační úlohy v internetovém prohlížeči. Uživatel, který bude s výukovým panelem pracovat, bude mít možnost se k němu přihlásit v rámci školní internetové sítě. V prohlížeči bude nastavovat vstupní parametry regulace a výstupem mu bude graf se zobrazenými hodnotami pohybu kuličky po odporové dráze. Tato kapitola je věnována popisu technologii HTML5, vytvoření vizualizačního programu a jeho obsluhy z pohledu uživatele.

V předložené práci je popsáno řešení takové, že existuje jeden program (executiva s jednou úlohou) pro simulaci a další executiva s jednou úlohou pro reálný proces. Podle toho, která z těchto exekutiv je do řídicího počítače nahrána, pak lze prohlížet příslušnou vizualizaci.

Řídicí program REX také umožňuje, aby jedna executiva byla spjata s více úlohami (tasky). Program lze tedy upravit tak, že v řídicím počítači poběží zároveň simulace (jeden task) i reálný proces (druhý task). Je tak možné vizualizovat jak simulaci, tak reálný proces.

## 10.1 Virtuální laboratoř

Virtuální laboratoř automatického řízení slouží uživatelům k přístupu na virtuální modely reálných úloh. Za pomoci vizualizace a simulace znázorňují virtuální modely chování reálných úloh pro přivedené vstupní signály. Současně mohou být zobrazovány průběhy všech signálů regulačního obvodu. Virtuální laboratoře mohou být přístupné všem uživatelům pomocí internetového prohlížeče.

Výhody takto vzdáleně ovládaných modelů pomocí webového prohlížeče jsou:

- přístup do laboratoře není omezen časem ani místem,
- uživatel nemusí mít k dispozici žádné pomůcky ani měřicí přístroje,
- měření na modelu lze neomezeně opakovat,
- uživatelé pracují s přístroji, které jsou reálné,
- naměřená data odpovídají datům reálným,
- kvalitní a rychlé zpracování naměřených dat
- je vyloučeno fyzické zranění s osob,
- práce a příprava z domova,
- jedná se o moderní přístup, který šetří čas i vyučujícím.

## 10.2 Technologie HTML5

HTML (HyperText Markup Language) je značkovací jazyk pro hypertext. Značkovací jazyk slouží pro obohacení textu o dodatečné informace (nejčastěji o významu, struktuře a způsobu zobrazování jednotlivých částí textu). Hypertext je způsob strukturování textu, který není lineární.

Obsahuje tzv. hyperlinky neboli odkazy. Taktéž odkazuje i na jiné informace v systému a umožňuje snadné publikování, údržbu a vyhledávání těchto informací.

Jazyk HTML se v kombinaci s protokolem HTTP (HyperText Transfer Protocol) a hypertextovými odkazy URL (Uniform Resource Locator) stal základem WWW (World Wide Web) nástroje. HTML je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.

HTML5 je specifikace jazyka HTML a v současné době je ve stádiu návrhu organizace W3C. HTML5 má oproti svým předchůdcům velkou spoustu inovací. Umožňuje přehrávat multimedia přímo ve webovém prohlížeči a vytvářet v něm aplikace, které fungují i bez připojení k Internetu. Zajímavou novinkou je geolokační rozhraní, které umožňuje získat uživatelům informace o jejich aktuální poloze pomocí dat z GPS, mobilních sítí, Wi-Fi a podobně. Od verze HTML4 se liší novými, zkrácenými a rychlejšími zápisy značek.

Technologii HTML5 se využívá i v diplomové práci. V době vytváření vizualizačního programu byl používán internetový prohlížeč Google Chrome verze 26.0.1410.64 a Mozilla Firefox 19.0.2, kteří ještě s prohlížečem Chromium, jako jediní tuto technologii podporovali.

## 10.3 Vytvoření vizualizačního programu

Pro vytvoření vizualizačního programu je potřeba vektorového editoru. Program Inkscape je v tomto případě dobrou volbou, protože nabízí úpravu vektorové grafiky a je ke stažení zadarmo. Tvorba obrázků probíhá za použití W3C standardu škálovatelné vektorové grafiky (SVG). Vektorová grafika je dána čarami a křivkami. To je rozdíl oproti bitmapové grafice, která je dána barevnými body. Takže v případě vektorové grafiky je možné libovolně měnit velikost výsledku, aniž by došlo ke zhoršení kvality.

V diplomové práci je pro vizualizaci simulovaného procesu použit PID a LQG regulátor. Pro vizualizaci reálného procesu se používá pouze LQG regulátor. Html soubory řídicího systému REX umožňují vytvářet vizualizace pro dvě prostředí. První prostředí se nazývá WebView a druhé WebSerVis:

- *WebSerVis* - je servisní vizualizace, která je určena k rychlému zobrazení dat z řídicího systému. Její vytvoření je jednoduché a intuitivní, avšak pro novou vizualizaci je potřeba upravit zdrojový kód stránky dle vizualizace z příkladu *mtuner* nebo *simple\_pid*. [6]
- *WebView* - je nástroj, který umožní poloautomatický převod schématu nakresleného pomocí vývojového nástroje RexDraw do HTML stránky. Díky tomu je možné zobrazit aktuální data z jednotlivých bloků. [6]

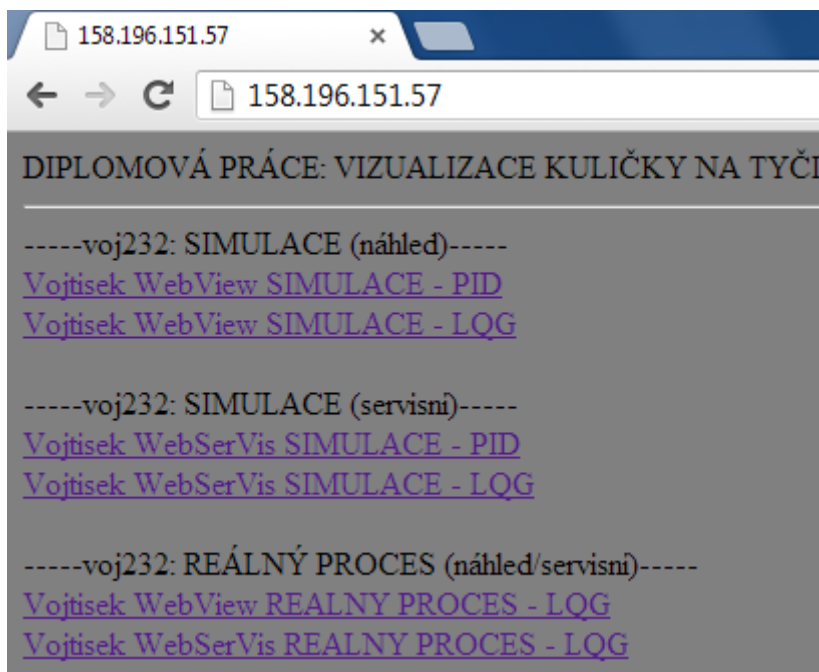
Vytvořený regulační obvod v REXu se exportuje do .svg souboru. Tento soubor se otevře v Inkscape a upraví pro vizualizaci. Takto upravený soubor bude uživatelem viditelný na internetové stránce, ze které bude úloha ovládána.

Nový upravený .svg soubor se otevře v textovém editoru WordPad nebo v programu PSPad a jeho obsah se nakopíruje do příslušného řádku souboru s příponou .html. Tento soubor je uložen v adresáři *REX/HTML/HMI/WebView* a slouží, jako vizualizace úlohy s regulačním blokem PIDU. Pokud má být vizualizace odzkoušena na osobním počítači s OS Windows, musí být nově vytvořený .html soubor uložen ve stejném adresáři jako původní.

Další vytvořená vizualizace je tzv. servisní. Opět byl převzat .html soubor z úlohy s regulačním blokem PIDU, který se nachází v adresáři *REX/HTML/HMI/WebSerVis*. Servisní vizualizace obsahuje místo obrázku regulačního obvodu jednotlivá okna se signály, které programátor navolí. Pod okny je umístěn graf trendu s průběhy signálů v čase. V době vytvoření diplomové práce nebyl graf trendu součástí instalačního souboru REX. Proto je přidán jako samostatný soubor v příloze IV.

Všechny vizualizace jsou založeny na HTML5 a komunikují pomocí technologie WebSocket. Aby bylo možné webové stránky propojit přímo s řídicím systémem je potřeba nejdříve aktivovat webový server *Lighttpd* a WebSocket server *RexWSTcp*. Oba servery se spouštějí pomocí souboru *launch-hmi.bat*. Odkaz na tento soubor je možné nalézt v menu *Start/RexControls/<aktuální verze rexu>/Nastroje/HMI/services*. [6]

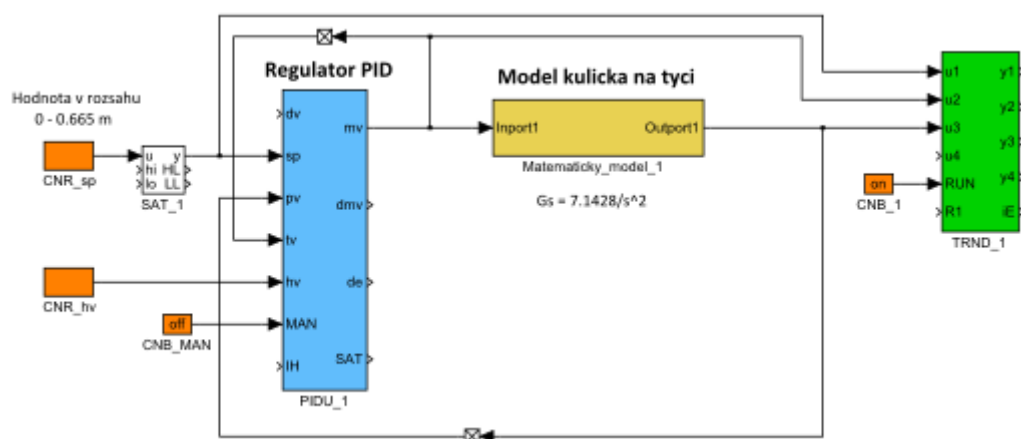
Vizualizační .html soubory jsou nahrány přes program Putty do řídicího počítače ALIX. Uživatel k nim přistupuje zadáním IP adresy ALIXu v prohlížeči (158.196.151.57) Obr. 10.3.



Obr. 10.3: Stránka s vizualizačními okny v prohlížeči Chrome

### 10.3.1 Vizualizace simulace

Výše byl popsán postup pro vytvoření jednotlivých vizualizačních obrazovek pro regulační obvod s PID a LQG regulátorem regulující matematický model kuličky na tyči. Obrázek regulačního obvodu na Obr. 10.3.1a je na stránce uložen pod názvem *Vojtisek WebView SIMULACE - PID*.



Obr. 10.3.1a: Vizualizace simulace s PID regulátorem

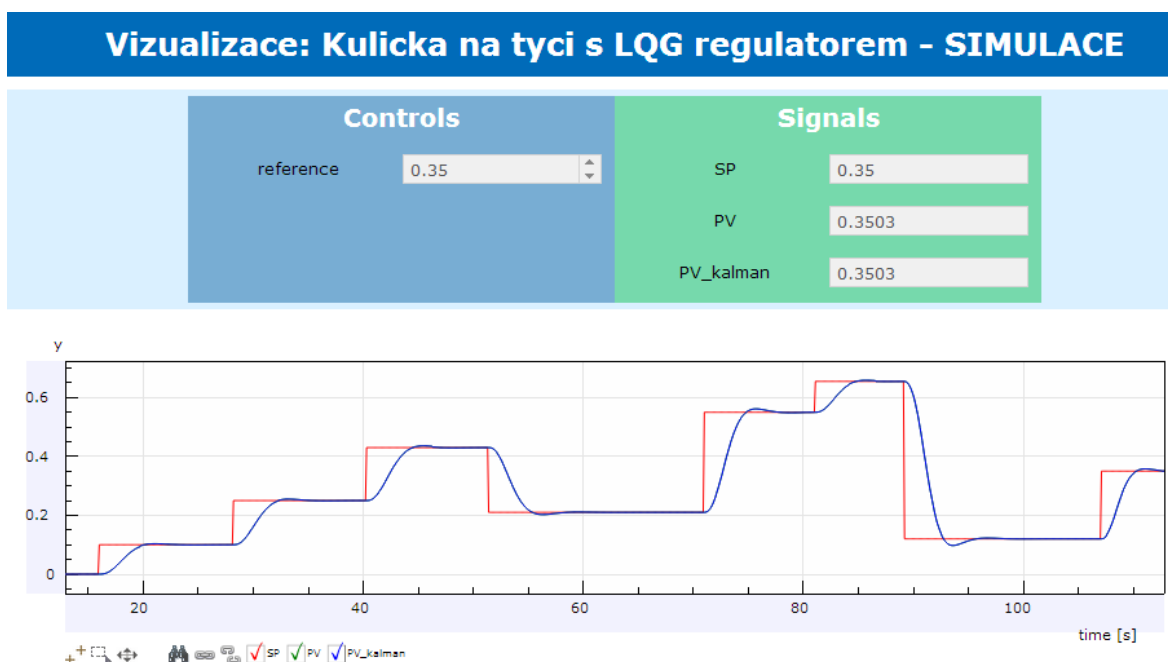
Na internetové stránce se načte obrazovka regulace s PID regulátorem a matematickým modelem kuličky na tyči. Uživatel nastavuje přes blok *CNR\_SP* polohu kuličky a může měnit parametry konstant regulátoru. Klepnutím na *TRND\_1* pravým tlačítkem se na vstupech zobrazí číselné hodnoty přicházející do trendu. To je možné sledovat i na vstupech a výstupech bloku *PIDU\_1*.

Další vizualizační obrazovka pro PID regulaci je v ALIXu uložena pod názvem *Vojtisek WebSerVis SIMULACE - PID* (Obr. 10.3.1b).





Obrázek 10.3.1c je součástí přílohy III, kde je zobrazen ve větším měřítku. K simulaci regulačního obvodu je také navržena servisní vizualizace, která je zobrazena na následujícím obrázku.



**Obr. 10.3.1d: Servisní vizualizace s LQG regulátorem**

V trendu je vidět průběh zadávání regulační veličiny a odezva regulované veličiny v čase. Regulátoru trvá přibližně 5s než působením akční veličiny na regulovanou soustavu dostane regulovanou veličinu na hodnotu reference. Akční veličina není v trendu zobrazována, protože její hodnota je v řádově tisících a v grafu by byla zobrazena jen jako přímka. Průběh regulačního děje je srovnatelný s průběhem v Simulinku.

### 10.3.2 Vizualizace reálného procesu

Na Obr. 10.3.2a je zobrazena vizualizace regulačního obvodu s reálnou soustavou. V bloku *Realny\_model\_1* se nachází bloky pro linearizaci v pracovním bodě a blok *Arduino\_UNO*, který představuje vývojovou platformu Arduino. V tomto bloku se nachází v/v piny.



## Vizualizace: Kulicka na tyči s LQG regulátorem - REALNY PROCES



**Obr. 10.3.2b: Servisní vizualizace reálného procesu**

Prodleva mezi změnou akční veličiny a změnou regulované veličiny je způsobena nedokonalým provedením kuličky, kdy je potřeba větší náklon (působit větším akčním zásahem) než by bylo s dokonale kulatou kuličkou potřebné.

## 11 Závěr

Hlavním úkolem diplomové práce bylo zkonstruovat fyzikální model kuličky na tyči a tento model řídit prostřednictvím programu vytvořeného v řídicím systému REX. Regulační program bylo nutné nahrát do zvoleného průmyslového PC. Za tímto účelem byl vybrán jednodeskový počítač ALIX, který je levným a kvalitním řešením.

V druhé kapitole jsem popsal vlastnosti servomotoru HS-311, který je v regulačním obvodu používán jako akční člen regulátoru. Uvedl jsem způsob řízení servomotoru a popsal princip PWM řízení, kterým je servomotor ovládán.

Počítač ALIX byl propojen s vývojovou platformou Arduino Uno, přes kterou se do regulačního obvodu přivádí signály PWM a čtou analogové signály nesoucí informaci o poloze kuličky na odporové dráze. Aby Arduino spolupracovalo s řídicím systémem REX, musel jsem do jeho procesoru nahrát ovladače zajišťující jejich vzájemnou kompatibilitu. V dalších fázích diplomové práce bylo zjištěno, že se musí upravit frekvence PWM pinů pro bezchybné (kontinuální) otáčení servomotoru, který se při provozu pod vahou fyzikálního modelu zasekával. Tento problém byl vyřešen přidáním několika řádků kódu do programu s ovladači. Vložený kód změnil hodnotu frekvence PWM výstupu pinu 5 a 6 na námi vhodnější frekvenci 61Hz.

V neposlední řadě jsem musel vytvořit samotný fyzikální model kuličky na tyči, který nemohl být příliš těžký, protože je zvedán malým modelářským servomotorem. Bylo nutné vyřešit konstrukční problémy s použitými materiály a upínání odporových drátů tak, aby se při napínání nepřetrhovali. Fyzikální model jsem posléze umístil na výukový stojan. Výukový stojan je vytvořen z hliníkových profilů a čelního panelu, na který jsem fyzikální model upevnil a do horní části panelu instaloval DIN lištu. Na této liště jsou umístěny zásuvky pro zdroje, svorkovnice pro signály, ALIX, Arduino Uno a krabička s elektronikou. Vytvoření elektroniky sebou přineslo několik problémů. Muselo být vyřešeno galvanické oddělení Arduina od zbylých částí elektroniky a nastavení zesílení signálu nesoucí informaci o poloze kuličky. V páté kapitole jsem dále popsal propojení jednotlivých zařízení pomocí jejich komunikačních rozhraní.

Identifikace modelu kuličky na tyči vycházela z analytických vztahů, které jsem převzal z materiálů uvedených v literatuře. Provedl jsem v programu MATLAB přepočty úhlu náklonu ramene na úhel pootočení servomotoru a získaný výsledek linearizoval v pracovním bodě tak, aby se dal použít v programu regulačního obvodu.

Po identifikaci a linearizaci v pracovním bodě jsem převzal z uvedené literatury soubory .m pro návrh LQG regulátoru. V diplomové práci je také uveden návrh PID regulátoru. Vypočítané hodnoty jsem implementoval do regulačního obvodu vytvořeného v prostředí Simulink pomocí funkčních bloků z knihovny RexLib. Následně jsem odzkoušel regulaci s použitím matematického modelu kuličky na tyči. V řídicím systému REX jsem za pomoci materiálu z uvedené literatury vytvořil regulační obvod pro regulátor PID a LQG. Regulátor PID pracuje pouze v simulaci (s matematickým modelem kuličky na tyči) a regulátor LQG byl implementován do simulačního regulačního obvodu i do regulačního obvodu s reálným procesem (fyzikálním modelem).

Součástí diplomové práce bylo pro vzniklé úlohy vytvořit vizualizační programy, které bude možno spouštět v internetovém prohlížeči a budou součástí tzv. virtuální laboratoře. Pro každý regulační obvod jsem za pomoci vzorových úloh navrhl dvě obrazovky. Jedna se v řídicím systému REX nazývá WebView a druhá WebSerVis. V těchto grafických prostředích uživatel

nastavuje parametry regulace a přehledně pozoruje dění v regulačním obvodu. V prostředí WebSerVis je provedeno měření výsledků zobrazovaných v grafu.

Do budoucna je možné diplomovou práci rozšířit o možnosti virtuální laboratoře. Studenti katedry by mohli k automatizačním úlohám přistupovat vzdáleně přes webový prohlížeč pomocí hesla. Virtuální laboratoř by logovala přihlášené uživatele a dal by se rezervovat čas pro práci s modelem.

# Literatura

- [1] OŽANA, Š. *Navrhování a realizace regulátorů*, Ostrava: VŠB-TUO, Ostrava 2012. Studijní materiály. ISBN 978-80-248-2605-9.
- [2] VOJČINÁK, P., PIEŠ, M., HÁJOVSKÝ, R. *Návrh LQG řízení pro fyzikální model kuličky na tyči*. Dokumentace k laboratornímu modelu VŠB-TUO, Ostrava 2012.
- [3] VOJTÍŠEK, R. *Návrh a realizace PID regulátoru pro řízení otáček stejnosměrného motoru na programovatelném automatu WinPAC-8000*. Bakalářská práce VŠB-TUO, Ostrava 2011.
- [4] ARDUINO. *ARDUINO* [online]. c2013. Dostupné z <<http://www.arduino.cc/en/Main/arduinoBoardUno>> .
- [5] RexControls. *Robust and Expert control systems* [online]. c2013. Dostupné z <<http://www.rexcontrols.com>> .
- [6] Firemní dokumentace Rex Controls.
- [7] Firemní dokumentace Matlab and Simulink (MathWorks).